# Analyzing Third Party Service Dependencies in Modern Web Services: Have We Learned from the Mirai-Dyn Incident?

Aqsa Kashaf
Carnegie Mellon University
akashaf@andrew.cmu.edu

Vyas Sekar
Carnegie Mellon University
vsekar@andrew.cmu.edu

Yuvraj Agarwal
Carnegie Mellon University
yuvraj@cs.cmu.edu

## Abstract

Many websites rely on third parties for services (e.g., DNS, CDN, etc.). However, it also exposes them to shared risks from attacks (e.g., Mirai DDoS attack [24]) or cascading failures (e.g., GlobalSign revocation error [21]). Motivated by such incidents, we analyze the prevalence and impact of third-party dependencies, focusing on three critical infrastructure services: DNS, CDN, and certificate revocation checking by CA. We analyze both direct (e.g., Twitter uses Dyn) and indirect (e.g., Netflix uses Symantec as CA which uses Verisign for DNS) dependencies. We also take two snapshots in 2016 and 2020 to understand how the dependencies evolved. Our key findings are: (1) 89% of the Alexa top-100K websites critically depend on third-party DNS, CDN, or CA providers i.e., if these providers go down, these websites could suffer service disruption; (2) the use of third-party services is concentrated, and the top-3 providers of CDN, DNS, or CA services can affect 50%-70% of the top-100K websites; (3) indirect dependencies amplify the impact of popular CDN and DNS providers by up to 25X; and (4) some third-party dependencies and concentration increased marginally between 2016 to 2020. Based on our findings, we derive key implications for different stakeholders in the web ecosystem.

## CCS Concepts

• **Security and privacy** → *Denial-of-service attacks*; • **Networks** → *Network measurement*; *Public Internet*; • **Computer systems organization** → *Redundancy*; *Availability*.

## Keywords

DDoS, redundancy, third-party dependency, DNS, CDN, OCSP

**ACM Reference Format:**
Aqsa Kashaf, Vyas Sekar, and Yuvraj Agarwal. 2020. Analyzing Third Party Service Dependencies in Modern Web Services: Have We Learned from the Mirai-Dyn Incident?. In *ACM Internet Measurement Conference (IMC '20), October 27–29, 2020, Virtual Event, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3419394.3423664

## 1 Introduction

Today, the web ecosystem has an increased reliance on third-party services such as DNS, CDN as also echoed in an IETF working group [28]. While many of these providers are well provisioned, history suggests that they are not entirely immune to failures; e.g., the Mirai Dyn attack [24], GlobalSign revocation error incident in 2016 [21] and the Amazon DNS DDoS attack in 2019 [50] affected a significant number of popular web services. These incidents raise broader questions about the robustness of the web ecosystem:

- Are these singular occurrences or are there other types of third-party services that are also potential Achilles' heels for affecting popular web-services? For example, as services are concentrated, is there a single provider whose failure will have a significant *impact* on many websites *critically dependent* on it?
- Are there hidden *transitive or indirect dependencies* between websites and their third-party providers concerning concentration and the extent of third party dependencies; e.g., loading `academia.edu` involves a third-party CDN MaxCDN, which in turn depends on AWS DNS!
- If, and how, did websites adapt, after the Dyn incident? Did they reduce their *critical dependency* on third party services? Did they become *redundantly provisioned* using multiple third-party providers for the same service?

We address these questions by carrying out a measurement study using Alexa's top-100K websites [49]. We focus on three infrastructural services that most modern websites critically rely on when servicing web requests: DNS, SSL certificate revocation checking by CAs, and content delivery (CDN). We analyze two kinds of dependencies: (1) *direct* dependencies such as the ones in the Dyn incident where a website like Spotify used Dyn as its DNS provider, and (2) *indirect* or transitive dependencies that consider "multi-hop" effects; e.g., loading *academia.edu* entails third-party CDN MaxCDN, which depends on AWS DNS. We take two snapshots in 2016 and 2020 to analyze trends in the dependency landscape of the web.

Our work is complementary to other concurrent efforts that study concentration in the web [28, 57]. First, we differ from other inter-website (e.g., JavaScript, fonts inclusion) dependency analysis [29, 35, 43, 47, 48, 62] as we study third-party dependency from an *infrastructure* standpoint. Second, compared to other efforts [9, 15, 55], we analyze both *direct and indirect* dependencies across websites and service providers, and find that these hidden indirect dependencies have significant impact. Third, we also do an evolution analysis by comparing two snapshots in 2016 and 2020 to highlight changes in the dependency landscape.

Our key findings are as follows:

- We show that 89% of the top-100K websites critically depend on third-party DNS, CDN, or CA providers, hence potentially compromising their availability.
- The use of third-party services is highly concentrated. Consequently, if the top-3 providers of CDN (Cloudflare, Incapsula, Cloudfront), DNS (Cloudflare, AWS DNS, DNSMadeEasy) or CA (Digicert, Let's Encrypt, Sectigo) services go down, then 50%-70% of the most popular websites will become unavailable.


*IMC '20, October 27–29, 2020, Virtual Event, USA*

ACM ISBN 978-1-4503-8138-3/20/10.
https://doi.org/10.1145/3419394.3423664

- Many service providers such as DNS, CDNs, and CAs critically depend on other third-party service providers. This critical dependence ranges from 17% to 35% across various inter-services dependencies. For instance, the largest CA DigiCert, critically depends on DNS provider, DNSMadeEasy. This dependency amplifies the impact of DNSMadeEasy from impacting 1% of websites directly to 25%.
- There is only a minor change in the dependencies of websites from 2016 to 2020 despite the highly publicized Dyn outage. In fact, critical dependency on third-party providers increased by 1% to 4.7%. Concentration also increased in DNS providers and CAs.

Our work has some limitations; e.g., we do not have capacity estimates for third-party services, we cannot infer third-party dependencies that are not visible to end hosts, etc. Our work is nonetheless a useful step towards establishing actionable metrics that can assist websites and service providers in making informed choices about their service dependencies. This in turn helps to mitigate the effects of large-scale incidents, improve resiliency, and minimize overall exposure to risk. Based on our findings, we derive implications for different stakeholders. Specifically, we recommend that: (1) web-services seek to increase their robustness by adding redundancy regarding the third-party services they use directly, while also determining the hidden dependencies (2) Third-party services should provide a quantitative understanding of their infrastructure and dependencies to the web services and should mitigate inter-service critical dependencies.

## 2 Motivation and Problem Scope

We discuss three motivating incidents that affected many websites and their users. We then define the types of dependencies we focus on to scope our analysis.

**Dyn DDoS Attack 2016:** In 2016, a DNS provider Dyn suffered a Distributed Denial of Service (DDoS) attack launched using the Mirai botnet. As a result, many popular sites were inaccessible for a few hours including Amazon, Netflix, Twitter, etc., since they used Dyn as their authoritative DNS provider. Furthermore, Fastly a content distribution network (CDN), also used Dyn as its authoritative nameserver. As a result, websites which did not use Dyn directly, but used Fastly, were also affected [24].

**GlobalSign Certificate Revocation Error 2016:** In 2016, the Online Certificate Status Protocol (OCSP) service of a certificate authority (CA) GlobalSign which provides the revocation status of a certificate, mistakenly marked valid certificates as revoked due to a misconfiguration [21]. This denied HTTPS access to many web-services e.g., Dropbox, the Guardian, and SoundCloud. This error persisted and affected websites for over a week, because of the caching of revocation responses. While caching may reduce impact of attacks on shorter time scales, but in incidents as mentioned above, it also extends the impact.

**Amazon Route 53 DDoS Attack 2019:** In 2019, Amazon's DNS service Route 53 suffered a DDoS attack lasting for 8 hours. As a result, other Amazon services such as S3, CloudFront, EC2, which relied on route 53 were also disrupted [50]. The attack also affected all the websites and service providers that used these Amazon services, e.g., Digital Ocean, a US-based cloud infrastructure provider.

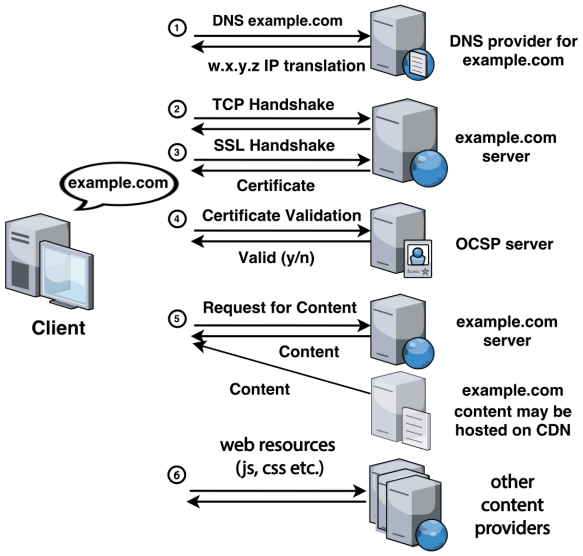



**Figure 1: The figure shows the life cycle of a web request and the different services it interacts with.**

Motivated by the impact of these incidents, and by the increasing popularity of third-party services, our goal is to analyze third-party dependencies of websites to assess the attack surface of modern websites and provide recommendations to increase their robustness.

### 2.1 Problem Scope

We look at the life cycle of a typical web request to highlight various services on its critical path illustrated in Figure 1. When a user requests a website, e.g. *example.com*, it is resolved to an IP address by contacting the authoritative nameserver (NS) of *example.com*. An NS (private or third party) of a website is the authority for the DNS records of that website. The request is then routed to the IP address of example.com's webserver. If the website uses HTTPS, it presents the client with its SSL certificate issued by a certificate authority (CA). The client additionally verifies the validity of the certificate by contacting servers that provide a Certificate Revocation List (CRL) called CRL distribution points or CDP, or uses the Online Certificate Status Protocol (OCSP) to ask the status of the certificate from an OCSP server. CDP and OCSP servers are managed by the certificate authority (CA). The addresses of CDPs and OCSP servers are included in the certificate. If the certificate is valid, the client requests the content of example.com, which may be hosted from a CDN. The website may also load content on its webpage from other content providers e.g., javascript libraries, fonts, etc. Our goal is to analyze dependencies with respect to *third-party providers* of services such as DNS, CDN, and certificate revocation checking by CAs. We do not analyze other dependencies on routing infrastructure or content providers. Regarding DNS, we do not look at DNS-over-TLS (DoT) and DNS-over-HTTPs (DoH) because of their currently low adoption [41].

### 2.2 Preliminaries

Before we formally define our measurement goals, we define actionable metrics that we use throughout our analysis. In the following,

consider a set of websites $W = \{w_1, w_2, ..., w_n\}$, and a set of services used by these websites $S = \{s_1, s_2, ..., s_m\}$. Let $P^s$ be a set of all providers of service type $s$ e.g., CDN, DNS and CA.

- *Third-party dependency:* This occurs if the website $w \in W$ is using a service from an entity different from itself. For instance, in the Dyn incident, Dyn is a third-party provider owned by Dyn Inc, and it serves websites for entities like Twitter *(twitter.com)* and Spotify *(spotify.com)*, etc.
- *Direct dependency:* This exists when a website $w \in W$ uses a provider $p \in P^{s_1}$ for getting the service $s_1$, e.g., in the Dyn incident, *twitter.com* had a direct dependency on Dyn, since it used Dyn as its authoritative nameserver. Similarly, a provider p' $\in P^{s_2}$ may also have a direct dependency on $p$ to get the service $s_1$; e.g., Fastly also used Dyn as its authoritative nameserver in 2016.
- *Indirect dependency:* This occurs when there is a direct dependency between a website $w \in W$ or a provider $p \in P^{s_1}$, and the provider $p' \in P^{s_2}$. Then if provider $p'$ has a direct dependency on another provider $p'' \in P^{s_3}$ for getting the service $s_3$, we say that $w$ or $p$ has an indirect dependency on $p''$. For example, in 2016 *pinterest.com* used Fastly CDN, while Fastly used Dyn for DNS services, leading to Pinterest being unreachable during the Dyn incident [20]. It can also exist in providers e.g., Certum CA uses MaxCDN which uses AWS DNS.
- *Critical Dependency:* When a website $w \in W$ or a service provider $p \in P^{s_1}$ uses another third-party provider p' $\in P^{s_2}$ for getting the service $s_2$, such that if p' is unavailable then service $s_2$ is denied to $w$ or $p$, then we say that $w$ or $p$ has a critical dependency on p' for service $s_2$. In the Dyn incident, *twitter.com* was critically dependent on Dyn for DNS. In contrast, if a website $w \in W$, or a service provider $p \in P^{s_1}$, uses multiple providers for service $s_2$ they are not critically dependent on any one provider and have *redundancy*. For example, *twitter.com* added redundancy by deploying a private DNS in addition to Dyn as we will see in Section 4.
- *Concentration of a service provider:* This counts the number of websites directly/indirectly dependent on a given provider. For example, if 100 websites use Dyn directly and 50 use it indirectly, we say that Dyn has a concentration of 150. Formally, let $D_w^p$ be the set of websites having direct dependencies on provider $p \in P^{s_1}$ and let $D_s^p$ be the set of all providers of service type $s \in S$ which have a direct dependency on $p$. Consider a function $f_c$ of $D_w^p$ and $D_s^p$, that gives the set of websites directly/indirectly dependent on $p$ then the concentration of the service provider $C_p$ is defined as:

$$C_p = \left| f_c(D_w^p, D_s^p) \right| = \left| D_w^p \cup \bigcup_{s=1}^{m} \bigcup_{k \in D_s^p} f_c(D_w^k, D_s^k \setminus \{p\}) \right|$$

  Here, to compute the concentration, we take a union of direct dependencies of provider $p$ with the direct dependencies of other providers which use $p$.
- *Impact of a service provider:* This counts the number of websites critically dependent on a service provider. For example, if 100 websites use Dyn, and 80 of them are critically dependent on it, then Dyn has an impact on 80 websites. Formally, let $E_w^p$ be the set of websites that are critically dependent on provider $p \in P^{s_1}$ and let $E_s^p$ be the set of all providers of service type $s \in S$ critically dependent on $p$. Consider the function $f_i$ of $E_w^p$ and $E_s^p$, that gives the set of websites critically dependent on $p$, then the impact of the service provider $I_p$ is defined as:

$$I_p = \left| f_i(E_w^p, E_s^p) \right| = \left| E_w^p \cup \bigcup_{s=1}^{m} \bigcup_{k \in E_s^p} f_i(E_w^k, E_s^k \setminus \{p\}) \right|$$

  Here, we consider the union of direct critical dependencies of provider $p$ with the critical dependencies of other providers which critically use $p$, to calculate its impact.

## 2.3 Research Questions

Given these actionable metrics, we can now concretely define our research questions :

- What fraction of websites have *critical dependencies* on third-party providers for DNS, CDN, and CA services?
- How concentrated is the web ecosystem with respect to *provider impact*? Are there single points of failure in the internet in terms of *provider impact*?
- What is the effect of *indirect dependencies* on the prevalence of third party dependencies, and *provider impact*?
- How has the world changed since the Dyn incident in terms of *critical dependency* of websites, *concentration* of service providers, and inter-service dependencies?

# 3 Methodology

In our study, we primarily focus on the Alexa top-100K websites. The Alexa list gives a good selection of functional websites frequently visited by users as observed in [53]. To analyze the current state of web dependencies, we use Alexa's January 2020 list. Table 1 summarizes our data. To study the change in dependencies after the Dyn incident, we use a snapshot of the rankings from December 2016 and collect data for these websites in 2016 and 2020 from the same vantage point to draw comparisons. Notably, 3.8% of the websites in the Alexa'16 list do not exist in 2020 and hence are excluded from our comparison analysis. Table 2 summarizes our data for the comparison analysis. We conduct our measurements from a single vantage point on the US East Coast.

Recall we are interested in measuring the third-party dependencies of websites on authoritative Domain Name Services, Content Delivery Networks, and Certificate Authorities for revocation information (OCSP servers and CRL distribution points). However, there is no ground truth on what constitutes a third-party; to this end, we consider two natural strawmen and describe our methodology to extend these to be more robust. We use the same techniques for the dataset based on Alexa's 2020 list, and for the comparison dataset based on Alexa's 2016 list.

## 3.1 DNS Measurements

**Identifying Third Party Nameservers:** Two approaches used in prior work are: (1) Matching TLDs with the nameservers [35] and (2) Matching the Start of Authority Records (SOA) of the nameservers and the website [7]. The SOA matching heuristic does not perform well because in many cases where a website uses a third party nameserver, the SOA for those websites also points to the

| | |
|---|---|
| Characterized websites for DNS analysis | 81,899 |
| Websites using CDNs | 33,137 |
| Characterized websites for CDN analysis | 33,137 |
| Websites supporting HTTPs | 78,387 |
| Characterized websites for CA analysis | 78,387 |

**Table 1: Summary of the websites we considered in our analysis for dependencies in 2020. Characterized websites are those for which we were able to establish if they use a third-party provider or a private one. We used the Alexa 2020 list for this analysis.**

| | |
|---|---|
| Characterized websites for DNS analysis | 87,348 |
| Websites using CDN either in 2016 or 2020 | 47,502 |
| Characterized websites for CDN analysis | 46,943 |
| Websites supporting HTTPs either in 2016 or 2020 | 69,725 |
| Characterized websites for CA analysis | 69,725 |

**Table 2: Summary of the total websites we considered in our comparison analysis for dependencies in 2016 vs. 2020. Characterized websites are those for which we were able to establish if they use a third party provider or a private one. We used the Alexa 2016 list for this analysis. 3.8% of the websites in the Alexa'16 list were not accessible in 2020.**

third-party DNS provider e.g., *twitter.com* SOA also points to Dyn since Twitter uses Dyn for DNS and thus we may incorrectly infer that Twitter is not using a third-party service leading to underestimation of third-party dependencies. On the other hand, the TLD heuristic works well in most cases but it misses some cases where providers use aliases. For instance, the nameserver for *youtube.com* is **.google.com*, which is the same logical entity and not a third-party, overestimating third-party dependencies. To avoid the pitfalls of both approaches, we develop a simple heuristic that combines TLD matching, SOA information, and other metadata to detect third-party providers more reliably.

We summarize our heuristic below:

$NS \leftarrow DIG\_NS(w)$
**for** $ns \in NS$ **do**
  $ns.type \leftarrow unknown$
  **if** $tld(ns) = tld(w)$ **then**
    $ns.type \leftarrow private$
  **else if** $isHTTPS(w)$ & $tld(ns) \in SAN(w)$ **then**
    $ns.type \leftarrow private$
  **else if** $SOA(ns) \neq SOA(w)$ **then**
    $ns.type \leftarrow third$
  **else if** $concentration(ns) \geq 50$ **then**
    $ns.type \leftarrow third$
  **end if**
**end for**

For all nameservers given a website, we first apply TLD matching. Then for the remaining (website, nameserver) pairs, we look at subject alternate names (SANs) present in the SSL certificate of websites, if they support HTTPS. All TLDs present in the SAN list of a website will also belong to the same logical entity. For example, **.google.com* will exist in the SAN list of *youtube.com* and hence, it will classify **.google.com* as private for *youtube.com*. For the remaining pairs, we fetch the SOA records (e.g., using `dig`) of the nameservers and the websites to look for mismatches, implying different DNS authorities. For instance, the SOA record for *amazon.com* is **.amazon.com* and its nameservers are **.dynect.net* (Dynect) and **.ultradns.net* (UltraDNS). The SOA records for Dynect and UltraDNS do not match to that of *amazon.com*, implying that amazon uses two third party DNS providers. Finally, we look at the concentration of a nameserver, which if large (e.g. > 50), implies a likely third-party provider.

We validate our heuristic using a random sample of 100 websites and manually verifying them. Our approach classifies (website, nameserver) pairs with 100% accuracy, while TLD and SOA matching classify with 97% and 56% accuracy respectively. We get 155,151 distinct (website, nameserver) pairs for the top-100K websites. 13.5% of these pairs remain uncharacterized. 18% of the top 100k websites appear in these pairs and we conservatively exclude them from our analysis. Rev C.7 is already mentioned here.

**Measuring Redundancy:** We need to identify not only if a website uses a third-party DNS provider, but also if it is redundantly provisioned. This means we need to identify that if $ns \in NS$ belong to different providers. For instance, **.alicdn.com* and **.alibabadns.com* belong to Alibaba, however, their TLDs are different. If a website has these nameservers, then it is not redundantly provisioned since they belong to the same entity. If two nameservers used by a website have the same TLD or the same SOA RNAME (administrator email address) or SOA MNAME (master nameserver address) records [31], we say that the nameservers belong to the same entity. For example, **.alibabadns.com* is the SOA MNAME for both **.alicdn.com* and **.alibabadns.com*.

## 3.2 Certificate Revocation Information

We extract the CRL distribution points (CDP), and OCSP server information from the SSL certificate of the website. Of the 100K websites, 78,387 support HTTPS. We observed 59 distinct CAs which provided CDPs and OCSP servers to our set of websites.

**Identifying Third Party CAs:** Certain private CAs issue certificates and provide revocation checking for their own domains only e.g., Google, Microsoft, etc. Since we focus on third party dependencies, we need to classify third party CAs. As is the case for DNS, simple TLD matching performs well but for some cases, it overestimates third party CAs, e.g., private CA Google Trust Services TLD *pki.goog* will not match many google domains such as *youtube.com*. To address this, we additionally use the SAN list and SOA information. We compare the SOA records of the CA address and the website where a mismatch implies two separate DNS authorities and we classify the CA as a third-party as shown below:

$ca \leftarrow getCA(w)$
$ca\_url \leftarrow getCA\_URL(ca, w)$
$ca.type \leftarrow unknown$
**if** $tld(ca\_url) = tld(w)$ **then**
  $ca.type \leftarrow private$
**else if** $isHTTPS(w)$ & $tld(ca\_url) \in SAN(w)$ **then**
  $ca.type \leftarrow private$
**else if** $SOA(ca\_url) \neq SOA(w)$ **then**
  $ca.type \leftarrow third$
**end if**

We validate our heuristic by taking a random sample of 100 websites and manually establishing their ground truth. We observed that our approach classifies (website, CA) pairs with 100% accuracy, while TLD and SOA matching classifies with 96% and 94% accuracy respectively.

**Measuring OCSP Stapling:** To see if a website has a critical dependency on OCSP responders and CDPs, we see if it has enabled OCSP Stapling because then the revocation status of the certificate comes stapled from the webserver. The user does not have to contact the OCSP server or CDP to get that information, thus eliminating its critical dependency on the CA. To measure OCSP stapling, we fetch the certificate for each website using OpenSSL [66]. An OCSP response stapled with the certificate implies support for OCSP stapling. Of the websites that support HTTPS, 28.5% support OCSP stapling.

## 3.3 CDN Measurements

We need to detect whether a website uses a Content Delivery Networks (CDNs). Most CDNs use CNAME (canonical name) redirects to point resources to the CDN e.g., *www.example.com* might point to *customer-1234.example-CDN-company.net*. Hence, one way to detect a CDN is to look at CNAME redirects for the *internal (website-owned)* resources of a website and match it against a *CNAME-to-CDN* map [10, 12]. Another way is to look at the autonomous system (AS) number of each *internal* resource and map the AS to popular CDNs [12, 42, 56]. The efficacy of both methods depends on the CNAME to the CDN mapping list in the first case, and AS to the CDN mapping list in the second case. We use the first method of CNAME redirects, which requires the identification of internal resources.

**Finding Internal Resources:** We fetch and render the landing page of the website using phantomJS, a headless browser [22], and record all hostnames that serve at least one object on the page. To identify internal resources, the baseline would be again to use TLD matching [35], which can reliably identify some internal resources, but also misses others, e.g., if *yahoo.com* loads an image from **.yimg.com*, which is an internal resource. Hence, we employ additional heuristics to identify internal resources, such as the subject alternate names (SAN) list in the SSL certificate of the website, public suffix lists [38, 65] and SOA records (if different SOA, then external). Next, we perform dig CNAME queries on all the internal resources of the webpage and extract the CDNs using our own self-populated CNAME-to-CDN map. We treat a provider as a CDN if it advertises itself as a CDN.Note that we do not determine which resources are essential to load the webpage of a website to determine critical dependency on CDNs. We only see if a website uses one or more CDNs to determine critical dependency. After this, we need to identify third party CDNs.

**Identifying Third Party CDNs:** A baseline to classify third-party CDNs would be to match the TLDs of the website and the CNAMES used by CDNs. This technique performs well but in some corner cases, it will have false positives, e.g., yahoo uses a private CDN which uses **.yimg.com* as CNAME. We can also match the SOA records of CNAMEs and websites, which also has false positives; e.g., Facebook CDN uses Facebook DNS as its SOA, Instagram uses private Facebook CDN while its SOA is AWS DNS. This technique also leads to overestimating third party CDNs. Hence, we develop a heuristic that uses TLD matching but to avoid some corner cases, it uses the SAN list and SOA information. For each (website, CDN) pair, we get the CNAMEs of the internal resources of the website which uses that CDN. For instance, the CNAME of the resources fetched from Akamai will contain CNAMEs such as **.akamaiedge.net*. For each CNAME, we first apply TLD matching, then we use the SAN list. Finally, we check for a mismatch in the SOA of the CNAME of the CDN and the SOA of the website, implying two separate entities. Of the 38,030 (website, CDN) pairs, we successfully classify 37,259 as third-party and we observe 86 distinct CDNs. We summarize our heuristic below:

$CDNS \leftarrow getCDN(w)$
**for** $cdn \in CDNS$ **do**
  $cnames \leftarrow getCNAMES(cdn, w)$
  $cdn.type \leftarrow unknown$
  **for** $cname \in CNAMES$ **do**
    **if** $tld(cname) = tld(w)$ **then**
      $cdn.type \leftarrow private$
    **else if** $isHTTPS(w)$ & $tld(cname) \in SAN(w)$ **then**
      $cdn.type \leftarrow private$
    **else if** $SOA(cname) \neq SOA(w)$ **then**
      $cdn.type \leftarrow third$
    **end if**
  **end for**
**end for**

Again, we validate our heuristic by taking a random sample of 100 websites and manually establishing their ground truth. We observed that our approach classifies (website, CDN) pairs with 100% accuracy, while TLD and SOA matching classify with 97% and 83% accuracy respectively.

## 3.4 Inter-services dependencies

DNS, CDN, and CA services also have dependencies on each other. For instance, CA's OCSP servers and CDPs, and CDNs will use DNS for IP resolution. CAs may also host CDPs or OCSP servers on a CDN. For a $CDN \rightarrow DNS$ dependency, we find the nameservers of the CNAMEs used by a given CDN and then classify them as a third party or private using the techniques mentioned in Section 3.1. We do the same for $CA \rightarrow DNS$ dependency by measuring and classifying OCSP and CRL addresses. Moreover, for $CA \rightarrow CDN$ dependency, we identify CDNs of OCSP servers and CDPs by getting their CNAMEs as we did for websites in Section 3.3. After identifying CDNs, we classify them as a third party or private using the same techniques described in Section 3.3.

## 3.5 Limitations

We acknowledge several limitations of our methodology to help put our results in perspective:

- We use a single vantage point. While this is a representative view of the structural dependencies, we may miss region-specific dependencies; e.g., websites in Asia having different dependency structure for clients in Asia, etc. Hence, our results might underestimate the impact of certain providers.
- We do not measure physical and network infrastructure dependencies; e.g., physical hosting, routing, or capacity. Such data is proprietary and hard to get in practice. Our analysis has value even with this limitation.
- We do not focus on dependencies between web-services themselves; e.g., loading third-party widget or scripts. While this can have implications (e.g., privacy), our focus is on infrastructure components like DNS, CDN, and CAs. We refer readers

to related work on the analysis of third-party web content (e.g., [7, 29, 35, 43, 47, 48, 51]).

- We only analyze dependencies on the landing pages. This is representative as shown by [35] where they found that on average, the root page of each site loads content from 87% of the union of external domains that all pages (landing and internal) depend on. However, we miss dependencies that may manifest deeper in the content hierarchy.

# 4 Direct Dependencies

In view of our research goals mentioned in Section 2.3, we analyze direct dependencies to 1) see how pervasive third party dependencies are, 2) look at the concentration of websites among third party providers and identify single points of failure in the internet, 3) compare the state of the third-party dependencies right after the Dyn 2016 attack, and now.

## 4.1 Third Party Dependencies

**Observation 1**: ***DNS third party and critical dependencies are higher for less popular websites. 89% of the top-100K websites use a third party DNS as compared to 49% in the top-100. Moreover, 28% of the websites are critically dependent in top-100 as compared to 85% in the top-100K.***

Figure 2 shows that third-party and critical dependencies are higher for lower-ranked websites. This could be because less popular websites cannot afford private infrastructure. Moreover, we observe that redundancy decreases with popularity; i.e., more popular websites care more about availability as compared to less popular ones. Overall, a very small fraction of websites have redundancy, perhaps since configuring for multiple providers can be non-trivial [4, 23]. Using multiple DNS providers requires provider support and currently, only a limited number of providers support or encourage redundancy as we see in Section 4.2.

**Observation 2**: ***Critical dependency on DNS providers has increased by 4.7% in 2020 relative to 2016.***

Table 3 shows that 6% of the top-100K websites that were critically dependent in 2016, have moved to a private DNS in 2020. On the other hand, 10.7% of the websites which used a private DNS in 2016, have moved to a single third party DNS provider (e.g., *espn.com, flickr.com*). Between these snapshots, redundancy has remained roughly similar. Overall, critical dependency has increased by 4.7% in 2020. More popular websites, however, have decreased their critical dependency.

**Observation 3**: ***Of the 33.2% websites using CDNs, more popular websites are less critically dependent on them. For the websites using CDNs, 85% (top-100K) and 43% (top-100) are critically dependent on a third-party CDN.***

We note that only 33.2% of the top-100K websites use CDNs. Figure 3 shows that 97.6% of websites that use CDNs use a third-party CDN. Third-party dependency increases across ranks; i.e., less popular websites likely cannot afford setting up a private CDN.

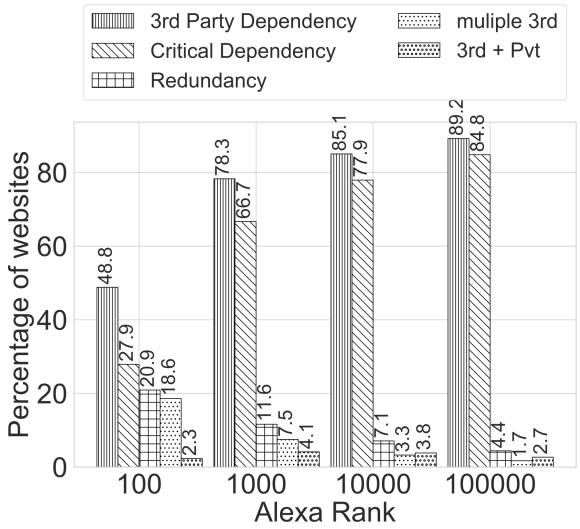



**Figure 2: Percentage of websites that use third-party DNS, are critically dependent, use multiple third-party DNS, and use redundancy by having private as well as third-party DNS providers. Critical dependency increases across ranks.**

| Website Trends | k=100 | k=1K | k=10K | k=100K |
|---|---|---|---|---|
| Pvt to Single 3rd | 0.0 | 7.4 | 9.8 | 10.7 |
| Single Third to Pvt | 1.0 | 1.6 | 4.2 | 6.0 |
| Red. to No Red. | 1.0 | 1.6 | 1.0 | 0.5 |
| No Red. to Red. | 2.0 | 1.9 | 1.1 | 0.5 |
| **Critical dependency** | **-2.0** | **+5.5** | **+5.5** | **+4.7** |

**Table 3: The percentage of websites per rank in $website \rightarrow DNS$ dependency trends in 2016 vs. 2020. Red. is short for redundancy. Critical dependency increased in 2020.**

Moreover, 85% of websites in this set (i.e., 28% of all sites) are critically dependent on a third-party CDN provider. We see that redundancy decreases with popularity; i.e., the top-100 websites use redundancy more often than the top-100K websites.

**Observation 4**: ***We observe no significant change in critical dependency on CDNs in 2020 relative to 2016.***

In 2016, 28.4% of the top 100K websites used CDNs in Alexa's 2016 rankings. In 2020, this number has increased to 39.9% for the same set of websites. Specifically, 18.6% additional websites have started using a CDN, while 6.8% have stopped using a CDN. Table 4 shows a rank-wise (as per the Alexa 2016 list) summary of the trends we observe in websites in 2016 vs. 2020. 0.5% of websites have moved to a single third-party CDN, while none have shifted to a private CDN. Moreover, 1.1% of websites have given up redundancy, including *twitch.tv, walmart.com, fiverr.com*. Also, 1.6% of the websites such as *paypal.com, imdb.com, ebay.com*, have adopted redundancy. Overall, 1.6% of websites have become critically dependent and 1.6% have become redundantly provisioned. Hence, we observe no significant change in third party dependency, critical dependency, and redundancy of top-100K websites. However, critical dependency has decreased in more popular websites.

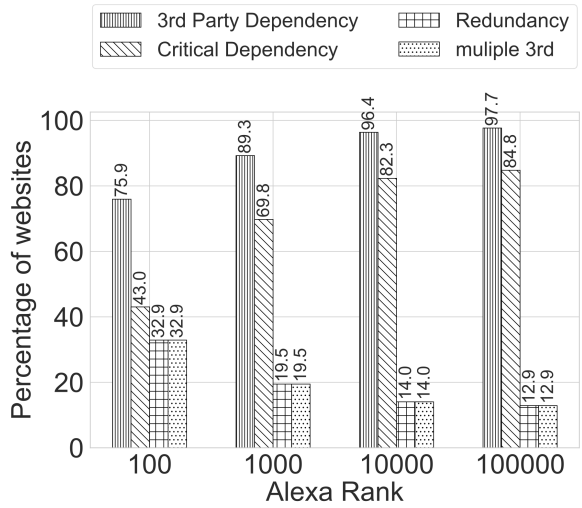



**Figure 3: Percentage of websites that use third-party CDNs, are critically dependent, use multiple CDNs to have redundancy. Critical dependency increases across ranks.**

| Website Trends | k=100 | k=1K | k=10K | k=100K |
|---|---|---|---|---|
| Pvt to Single 3rd party CDN | 0.0 | 0.3 | 0.8 | 0.5 |
| 3rd Party CDN to Pvt | 0.0 | 0.0 | 0.0 | 0.0 |
| Red. to No Red. | 3.0 | 2.7 | 1.2 | 1.1 |
| No Red. to Red. | 9 | 6.8 | 3.0 | 1.6 |
| **Critical dependency** | **-6.0** | **-3.8** | **-1.0** | **+0.0** |

**Table 4: Percentage of websites per rank in $website \rightarrow CDN$ dependency trends in 2016 vs. 2020. Red. is short for redundancy. We observe no significant change between 2016-2020.**

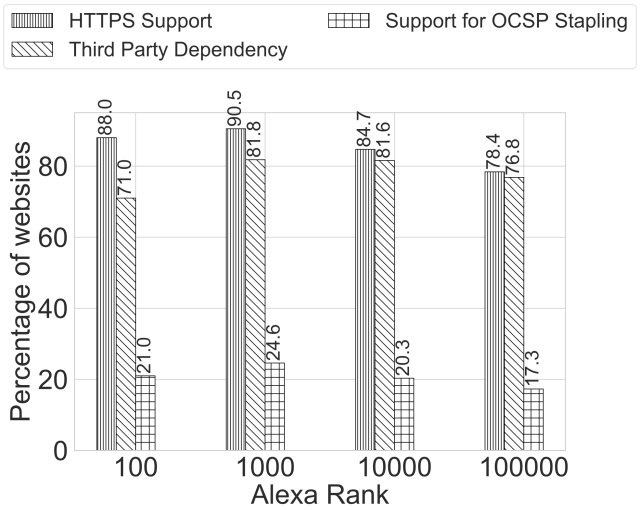



**Figure 4: Percentage of websites that use HTTPS, use third-party CA providers, and enable OCSP stapling shown per rank and hence are not critically dependent. Critical dependency is slightly higher in less popular sites.**

| Website Trends | k=100 | k=1K | k=10K | k=100K |
|---|---|---|---|---|
| Stapling to No Stapling | 7.5 | 6.2 | 9.1 | 9.7 |
| No Stapling to Stapling | 3.7 | 14.7 | 12.9 | 9.9 |
| **Critical dependency** | **+3.8** | **-8.5** | **-3.8** | **-0.2** |

**Table 5: The percentage of websites in $websites \rightarrow CA$ dependency trends in 2016 vs. 2020. We observe no significant change in critical dependency.**

**Observation 5**: ***More popular websites are slightly less critically dependent on third-party CAs.***

Our data indicate that 78% of the top-100K websites support HTTPS. 77% use a third-party CA, and 60% are critically dependent on the CA. Figure 4 shows the percentage of websites that support HTTPS across ranks, which is marginally higher for more popular websites. This may be because popular websites care more about user trust, due to a larger user base. The use of third party CAs is also higher (77% in top-100K) in less popular websites as compared to more popular (71% in top-100) websites. Critical dependency in CAs (support for OCSP stapling) remains low across all ranks as compared to DNS and CDNs, being only 17% for the top-100K websites. Hence, 61% of the top-100K websites are critically dependent on CAs as they do not support OCSP stapling. Low support of OCSP stapling may be due to poor/faulty support across browsers and web servers (Apache and Nginx) [6, 14]. Also, OCSP stapling is ineffective unless the browser knows when to expect a stapled response, as an attacker can just omit OCSP status when using a revoked certificate causing a soft-fail [14]. The must-staple extension addresses this but has yet to gain widespread support [14].

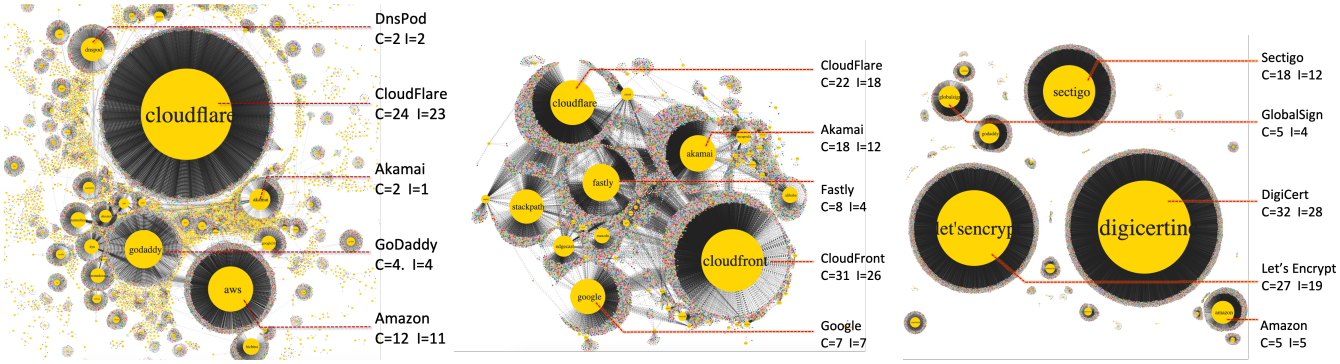
**Observation 6**: ***We observe no significant change in critical dependency of websites on CAs in 2016 vs. 2020.***

In 2016, 46,529 websites from Alexa's 2016 list of top-100K supported HTTPS, which has increased to 69,725 websites in 2020. 23,196 additional websites have adopted HTTPS, and 11.9% of these support OCSP Stapling in 2020. 5% of these which supported OCSP stapling in 2016, existed in the top-100 including *haosou.com, naver.com*, etc.

9.7% of the websites that supported HTTPS in 2016, have dropping support for OCSP Stapling in 2020 as shown in Table 5, including popular websites such as *dropbox.com, wordpress.com, microsoft.com*, etc. This trend increases across ranks, 7.5% of websites in the top 100 as compared to 9.7% in the top 100K. We also observe that 9.9% of the websites that supported HTTPS but did not have OCSP stapling in 2016, have enabled OCSP stapling in 2020. Overall in terms of critical dependence, we observe no significant change in top-100K websites. However, in more popular websites we see an increase in critical dependency.

## 4.2 Provider Concentration

**Observation 7**: ***4 (out of 10K) DNS providers critically serve 50% of the top-100K websites, 2 (out of 86) CDNs critically serve 50% of the websites that use CDNs, and 2 (out of 59) CAs are critical dependencies for 50% of the websites that support HTTPS.***




(a) (b) (c)

**Figure 5: Figure 5a, 5b and 5c show the dependency graph of the top-100K websites and third-party DNS, CDNs, and CAs respectively. The size of a node is proportional to its indegree. We also label the *concentration C* and *impact I* of the top 5 providers in terms of the percentage of total websites. A few DNS, CDN and CA providers serve a large number of websites.**

At a high level, these results provide empirical confirmation of observations and concerns in the community [28, 57].

Figure 5a shows the concentration and impact as defined in Section 2.2, of major third-party DNS providers for the top 100K websites, in terms of the percentage of websites. DNS ecosystem is heavily concentrated, with just one DNS provider covering 23% and the Top-3 DNS providers critically serving ≈ 40% of the top 100K websites.

However, there are subtle differences across popularity ranges. For example, in Figure 5a the impact of CloudFlare shows that it critically serves 23% of the websites in the top 100K. Although we do not show here, it is not a major provider for more popular websites. Dyn is the most popular in the top-100 websites with 17% of the websites using Dyn and only 2% critically dependent on it.

Moreover, the difference in concentration and impact in Figure 5a shows the degree of redundancy for the websites using a particular provider. For instance, for CloudFlare, it is 1% (C-I = 24-23) of websites. The near-complete lack of redundancy in CloudFlare's consumers is because it requires that DNS traffic is routed through the CloudFlare network to protect against DDoS and other attacks. This approach does not allow domains to register a secondary DNS provider. Although we do not show here, we observe a higher degree of redundancy in the consumers of Dyn, NS1, UltraDNS, and DNSMadeEasy. This may be because these providers encourage the use of secondary DNS provider by giving specific guidelines to seamlessly incorporate a secondary DNS provider as also observed by [4]. High redundancy for Dyn and NS1 customers could also be a result of large-scale attacks on Dyn [24] and NS1 [5] also independently observed by [1].

Figure 5b shows the major third-party CDN providers in the top 100K websites. Our characterization of the top providers is based on their concentration defined in Section 2 and not on how much traffic they carry. We observe that the CDN market is also heavily concentrated; Amazon CloudFront supports 30% of the websites that use a CDN and the top 3 cover 56% of the websites that use a CDN which is 18.6% of the top-100K websites. Amazon CloudFront covers 30% of the websites in the top 100K that use a CDN. However, Akamai which covers 18% of the top 100K websites is more dominant in popular (top 100) websites as compared to Amazon CloudFront.

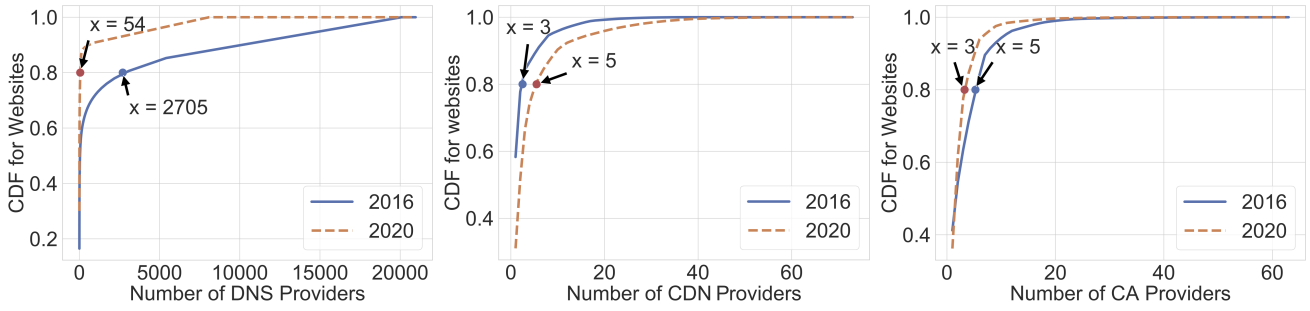
Moreover, if we see redundancy per provider by subtracting impact from concentration, we observe that very few websites using CloudFront and CloudFlare are redundantly provisioned, as compared to customers of Akamai or Fastly. We find that among top providers, Akamai and Fastly support multi-CDN strategy and provide specific guidelines to enable it [58, 59]. However, unlike DNS, using multiple CDNs does not always require provider support; e.g., when using CDN brokers [44]. As a result, there is a comparatively higher degree of redundancy in the consumers of CDN providers in contrast to DNS providers.

Finally, Figure 5c shows that there is also a significant concentration among CAs. 60% of the websites that support HTTPS (46.25% of the top 100K websites) in the top-100K websites are critically dependent on the top 3 CAs. DigiCert covers 32% of the top-100K websites and 44% (which is not shown here) of the top 100 websites that support HTTPS. Hence, it is equally popular across all ranks of websites. In terms of OCSP stapling support per provider, we observe that websites using Let's Encrypt and Sectigo have higher support for OCSP stapling as compared to other top providers like DigiCert, Amazon, and GlobalSign.

**Observation 8**: ***Between 2016 to 2020, concentration in DNS and CA providers has increased. However, concentration in CDN providers has decreased.***

Figure 6a shows that concentration in DNS providers has increased as 54 providers serve 80% of the websites in 2020 as compared to 2705 providers in 2016. The top 3 providers impacted 29.3% of the top-100K websites in 2016. In contrast, in 2020, the top 3 DNS providers impact 40% of the top-100K websites. The set of major DNS providers is roughly the same; in 2016 and 2020 the top 3 providers are CloudFlare, AWS DNS, and GoDaddy. Following the Dyn incident, we also observe a reduced footprint of Dyn; i.e., the concentration of Dyn has decreased from serving 2% of top-100K websites in 2016 to serving 0.6% of top-100K websites in 2020.

Figure 6b shows that unlike DNS, concentration in CDNs has marginally decreased in 2020. 5 CDNs serve 80% of the top-100K




(a) (b) (c)

**Figure 6: The CDF of websites against the number of DNS providers can be seen in Figure 6a for 2016 and 2020, against the number of CDNs is shown in Figure 6b, and against the number of CAs in Figure 6c. Between 2016-2020, concentration has increased in DNS and CA providers.**

| Dependency | Total | 3rd-Party Dep. | Critical Dependency |
|---|---|---|---|
| $CDN \rightarrow DNS$ | 86 | 31 (36%) | 15 (17.4%) |
| $CA \rightarrow DNS$ | 59 | 27 (48.3%) | 18 (30.5%) |
| $CA \rightarrow CDN$ | 59 | 21 (35.5%) | 21 (35.5%) |

**Table 6: For each dependency type, the table shows the number of total dependencies, third party dependencies and critical dependencies. Critical dependencies are also prevalent in service providers.**

websites in 2020 as compared to 3 CDNs in 2016. It means that the single points of failures in 2020 are smaller as compared to 2016. The top 3 CDNs in 2020 cover 18.6% of websites, as compared to 20.8% of websites in 2020. The set of popular providers has some minor churn; e.g., while CloudFlare was the top CDN provider in 2016, in 2020, Cloudfront has become the most popular provider as its impact increased by 6% in 2020.

Figure 6c shows that concentration in CAs has increased in 2020, as compared to 2016 as 3 CAs serve 80% of the websites in 2020 as compared to 5 CAs in 2016. While the Top 3 CAs in 2016 impacted 26% of the top 100K websites, in 2020, the top 3 CAs have an impact on 46.25% of websites. The top 3 providers have also changed from 2016 to 2020. Symantec which was the third most popular CA in 2016, has dropped off the top-3 list in 2020.[1] Let's Encrypt has become the second most popular provider in 2020, with an increase in impact from 2.4% in 2016 to 15% in 2020. Sectigo, formerly Comodo, was the top provider in 2016, and we see a decrease in its impact from 15% in 2016 to 9% in 2020.

## 5 Indirect Dependencies

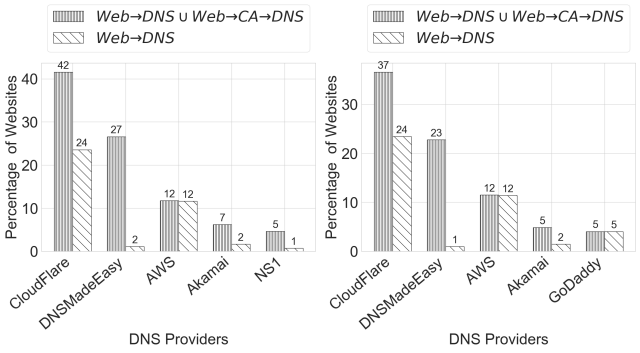
In this section, we look at inter-service dependencies such as CDN to DNS, or CA to DNS dependency. We analyze the following: 1) Are critical dependencies present among service providers, as they are for websites? 2) How do the indirect dependencies (defined in Section 2.2) arising from critical inter-service dependencies affect the third-party dependency of websites, and the concentration among providers?

### 5.1 $CA \rightarrow DNS$ Dependency

We find that out of the 59 CAs, 27 (48.3%) use a third-party DNS provider. Of these 27 CAs, 18 (66.67%) use a third-party DNS provider exclusively including top-3 CA providers DigiCert, Let's Encrypt, and GlobalSign. The use of specific 3rd party DNS providers by these 18 CAs is less concentrated: 4 use Comodo DNS, 3 use Akamai, and 3 use AWS DNS.

We find that three additional websites are dependent on a third party as they use a private CA, which itself uses a third-party DNS. These include *godaddy.com*, which uses GoDaddy CA but the CA uses Akamai as DNS, *trustwave.com*, and *wisekey.com* which are relatively lower ranked websites.

[1]Symantec CA business was bought by DigiCert [16]




(a) (b)

**Figure 7: Figure 7a and 7b show the top-5 3rd party DNS providers in terms of concentration and impact respectively in the top-100K websites when we consider $CA \rightarrow DNS$ and when we just consider $Web \rightarrow DNS$ dependency. Indirect dependencies amplify provider concentration and impact.**

**Observation 9**: ***72% of the websites are critically dependent on 3 DNS providers when we consider direct*** $CA \rightarrow DNS$ ***dependency as compared to 40% when we just account for*** $website \rightarrow DNS$ ***dependency***

Figure 7a shows that the concentration of providers has increased when we also consider $CA \rightarrow DNS$ dependency; e.g., in the case of CloudFlare by 18%, as it serves Let's Encrypt which is the second major CA. Similarly, for DNSMadeEasy the increase is from 2% to 27% of websites since it is used by DigiCert which itself serves 32% of the top-100K websites. Figure 7b highlights the change in impact of providers. Overall, 72% of the websites are critically dependent

| | |
|---|---|
| Private to Single Third Party | 1 (1.4%) |
| Single Third Party to Private | 9 (12.8%) |
| Redundancy to No Redundancy | 2 (2.8%) |
| No Redundancy to Redundancy | 0 (0.0%) |
| **Critical dependency** | **-6 (-8.6%)** |

**Table 7: Trends in $CA \rightarrow DNS$ dependency in 2016 vs. 2020. Critical dependency decreased in 2020.**

on just 3 DNS providers as compared to 40% of websites when we just consider direct $website \rightarrow DNS$ dependencies.

**2016 vs. 2020:** We observe a decrease in the critical dependency of CAs in 2020. Out of the 70 CAs in 2016 data, 33 (47%) used a third-party DNS provider and 24 (34.2%) were critically dependent on it. In 2020, we find that 9 of these critically dependent CAs (GeoTrust, Symantec, etc.) have moved to a private DNS, and 1 CA (Trust Asia) has moved to a single third party DNS provider from using a private DNS as shown in Table 7. Moreover, 2 redundantly provisioned CAs (Digicert, Internet2) have moved to a single third-party DNS. Overall, critical dependency has decreased by 8.6% because 9 critically dependent CAs in 2016 have shifted to a private DNS in 2020, while 3 have became critically dependent.

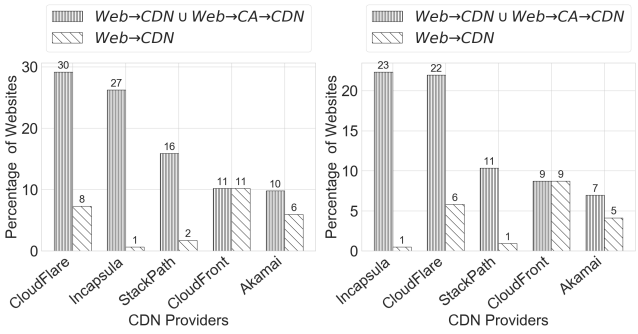
## 5.2 $CA \rightarrow CDN$ Dependency

Out of the 59 CAs that we observe, we find that 24 (40.6%) use CDNs, and 21(35.6%) use a third-party CDN and use it exclusively. These include major CAs such as DigiCert, Let's Encrypt, Sectigo, GlobalSign, etc. The critically dependent CAs cover 73.8% of the websites using HTTPs. Akamai and CloudFlare are the dominant CDNs that are used by 5 CAs each. As a result of this dependency, 32 additional websites now have a third-party dependency. Even though they use a private CA, it in turn uses a third-party CDN. This set includes many popular websites such as *microsoft.com, godaddy.com, xbox.com.*

**Observation 10**: ***56% of the websites are critically dependent on 3 CDNs when we consider*** $CA \rightarrow CDN$ ***dependency as compared to 18% when we only consider*** $website \rightarrow CDN$ ***dependency***

Figure 8a shows that the concentration of CDNs has increased when we also consider $CA \rightarrow CDN$ dependency. For instance, Cloudflare now covers 30% of the top-100K websites as compared to 7% when we just consider $website \rightarrow CDN$ dependency, because it is used by Let's Encrypt which is the second major CA. Similarly, the concentration of Incapsula has increased from 1% of websites to 27%, as it serves DigiCert which is used by 32% of the sites. Similarly, Stackpath serves Sectigo which is the third major CA, and hence we see an increase in its concentration from 2% to 16%. Figure 8b shows that the impact of CDNs also increases significantly. Overall, the top 3 CDNs in terms of impact have changed, previously the top 3 CDNs covered 18% of the websites and now the top 3 CDNs critically serve 56% of the websites.

**2016 vs. 2020:** We observe 70 distinct CAs in 2016 data of which 21(30%) used a CDN in 2016. Of these, 18 (25.7%) CAs used a third-party CDN exclusively including major CAs such as GeoTrust, GlobalSign, Symantec, GoDaddy. In 2020, TeliaSonera CA has moved from a third-party CDN to private in 2020, while three CAs have




(a) (b)

**Figure 8: The Figure 8a and 8b show the top-5 3rd party CDNs in terms of concentration and impact respectively in the top 100K websites when we consider $CA \rightarrow CDN$ and when we just consider $Web \rightarrow CDN$ dependency. Indirect dependencies amplify provider concentration and impact.**

| | |
|---|---|
| No CDN to Third Party CDN | 3 (14.28%) |
| Third Party CDN to no CDN | 2 (9.5%) |
| Private to Third Party | 0 (0.0%) |
| Single Third Party to Private | 1 (4.76%) |
| **Critical dependency** | **0 (+0%)** |

**Table 8: Trends in $CA \rightarrow CDN$ dependency in 2016 vs. 2020. There is no significant change in critical dependency of CAs.**

moved from having no CDN in 2016 to having a third party CDN in 2020 including a major CA Let's Encrypt as shown in Table 8. However, 2 CAs have also moved from having third-party CDNs to not having CDNs in 2020. Hence, third party and critical dependency has remained overall unchanged.

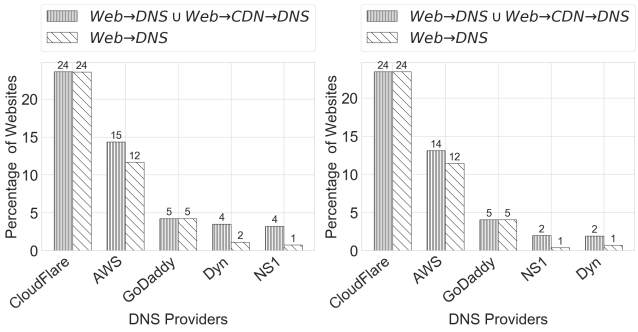
## 5.3 $CDN \longrightarrow DNS$ Dependency

We observe 86 CDNs in total, out of which 31 (36%) use a third party DNS provider and 15(17.4%) of them are critically dependent as shown in Table 6. However, the critically dependent ones are not significant CDN providers as they support only 1.5% of the top-100K websites using CDNs.

We find that 290 additional websites are now dependent on a third party as they use a private CDN, which in turn uses a third-party DNS provider. These websites include *twitter.com, airbnb.com, and squarespace.com.*

**Observation 11**: ***Major CDN providers use private DNS, hence we see little to no change in the impact of DNS providers as a result of*** $CDN \rightarrow DNS$ ***dependency.***

Figure 9a and 9b show the change in concentration and impact of the top 5 DNS providers respectively when we also consider $CDN \rightarrow DNS$ dependency. There is no significant change in the concentration of major providers as they use a private DNS. Only Fastly in the top 5 CDN providers, uses a third-party DNS provider Dyn. We observed that AWS DNS serves 16 of the CDNs, 7 of which use AWS exclusively. However, these 7 CDNs serve only 2% of the top-100K websites using a CDN.




(a) (b)

**Figure 9: The Figure 9a and 9b show the top-5 3rd party DNS providers in terms of concentration and impact respectively in the top 100K websites when we consider $CDN \rightarrow DNS$ and when we just consider $Web \rightarrow DNS$ dependency. Indirect dependencies amplify provider concentration and impact.**

| | |
|---|---|
| Pvt to Single Third Party | 0 (0.0%) |
| Single Third Party to Private | 1 (2.1%) |
| Redundancy to No Redundancy | 1 (2.1%) |
| No Redundancy to Redundancy | 2 (4.25%) |
| **Critical dependency** | **-2 (-4.25%)** |

**Table 9: Trends in $CDN \rightarrow DNS$ dependency in 2016 vs. 2020. Critical dependency has decreased in 2020 relative to 2016.**

**2016 vs. 2020:** We observe 47 distinct CDNs in the 2016 data. Of these 47 CDNs, 12 (25.5%) used third party DNS providers in 2016. 8 (17%) were critically dependent on a single third-party DNS provider in 2016. Of those 8 CDNs, 2 (Netlify, Kinx CDN) have adopted redundancy in 2020, and 1 (GoCache) has moved to a private DNS as shown in Table 9. However, 1 redundantly provisioned CA (Zenedge) in 2016 has moved to a single third-party CDN in 2020. Hence, critical dependency has decreased by 4.25%.

# 6 Additional Case Studies

The previous section considers the Alexa top-100K websites. However, this tangled web of dependencies extends into other domains. We present two example case studies of specific market verticals that highlight the impact of third party dependencies outside of popular websites.

## 6.1 Case Study I - Hospitals

Internet outages for hospitals can hamper hospital operations e.g., electronic health record (EHR) system hosted on a remote server, electronic transfer of prescriptions, and emails, etc. In fact, on March 22, 2020 a DDoS attack targeted a Paris hospital authority AP-HP, which manages 39 public hospitals, during the coronavirus pandemic [11]. Also, in the 2016 Dyn attack, AthenaHealth and AllScripts also suffered outage [40] since they used Dyn.

We analyze third-party DNS, CDN and CA dependencies in the top 200 US hospitals [46]. Table 10 shows the percentage of hospitals that use third party DNS, CDN, and CA and that are critically dependent. 51% of the hospitals use a third party DNS and only 6% have redundancy. GoDaddy DNS is the most concentrated provider which covers 13% of these hospitals. We observe that third party

| Service | Third-Party Dependency | Critical Dependency |
|---|---|---|
| DNS | 102 (51%) | 92 (46%) |
| CDN | 32 (16%) | 32 (16%) |
| CA | 200 (100%) | 156 (78%) |

**Table 10: Trends in third-party dependency of top 200 US hospitals. Hospitals are as critically dependent as websites.**

| Service | 3rd-Party Dep. | Redundancy | Critical Dependency |
|---|---|---|---|
| DNS | 21 (91.3%) | 1 (4.43%) | 8 (34.7%) |
| Cloud | 15 (65.2%) | 0 (0%) | 5 (21.7%) |

**Table 11: Trends in third party dependency of top smart home companies. Critical dependency is also prevalent in smart home services.**

DNS dependency is less prevalent (51%) in hospitals as compared to Alexa websites (89%). However, redundancy is equally rare, 90% hospitals that use a third party DNS are not redundantly provisioned, similar to the Alexa top-100K websites (95%). We observe that 16% of the hospitals use CDNs and all of them use a third-party CDN. CDN usage is less (16%) in hospitals as compared to Alexa websites (33.2%). Critical dependency on CDNs is 100% similar to Alexa websites (85%). Akamai is the most concentrated CDN provider which covers 7% of the hospitals. We notice that all hospitals support HTTPS and 22% of the hospitals support OCSP stapling. This is slightly more than Alexa sites, where 17% support OCSP stapling.

To summarize, third-party dependency is less for hospital websites relative to Alexa websites. However, trends in critical dependency are similar.

## 6.2 Case Study II - Smart Home Companies

Outage of smart home services can have serious consequences [54, 61], e.g., in 2017, an Amazon S3 outage caused many smart home devices (locks, lights, etc.) to not function [25].

Hence, we analyze 23 smart home companies for third-party dependencies, including smart home frameworks like Samsung SmartThings, Yonomi, Amazon, etc., and smart home devices such as Lifx, Philips Hue, etc. Of these 23 companies, 14 operate locally and on the cloud while 9 smart home companies are cloud-only.

Table 11 shows the trends we observe in smart home companies concerning third-party dependencies. Only 3 companies (Philips Hue, Apple Homekit, and Amazon Alexa) use a private DNS and only 1 company uses redundancy. Of the remaining 21, 13 have local fail-over and hence 8 are critically dependent including Logitech Harmony, Yonomi, Brilliant Tech, IFTTT, Petnet, Ecobee, Ring Security, etc. These trends of third party and critical dependency on DNS providers are similar to Alexa websites.

In terms of cloud usage, all companies use the cloud, while 15 use third-party cloud, and none of them are redundantly provisioned. Of these 15, only 5 lack local fail-over, and hence they are critically dependent on their cloud provider. These include Petnet, IFTTT, Logitech Harmony, Ecobee, and Ring Security. Moreover, we observe that 11 of the 15 smart home companies that use a third party cloud, use Amazon as their cloud provider, while 13 use Amazon DNS. All in all, third party and critical dependency have similar trends in smart home companies as in Alexa websites.

## 7 Related Work

**Content Dependency measurements:** Prior work has analyzed the third party content-level dependencies of websites. Kumar et al., [35] study HTTPS adoption and Podins et al., [48] measure the implementation of Content Security Policy, among third-party web content. Other efforts (e.g., [43, 47]) analyze malicious third party content such as JavaScript or fonts. Ruohonen et al., looks at the prevalence of cross-domain TCP connections [52]. Ikram et al., look at dependency chains in loading third party web content [29], while Urban et al., study how content dependencies change with repeated visits [62]. Our focus is on infrastructural dependencies and not on third party content.

**Dependency and Concentration Analysis:** Simeonovski et al., [55] analyze global scale threats where bad actors can be a country, an autonomous system or a service provider such as DNS or an Email server. Closer in spirit to our work, Allman analyzed DNS robustness of websites and makes a case for adding redundancy [3]. Other work looked at the behavior of Dyn and NS1 clients to make a case for DNS redundnacy [1]. NSDMiner discovers network service dependencies such as ISPs, from passively observed network traffic [45]. Other work looks at the dependency of websites on hosting providers [60] and the dependency of government websites on CAs and content providers [27]. Hoang et al., study web co-location using DNS measurements [26]. Similarly, Dell et al., analyze third party infrastructural dependencies of websites from a large scale DNS dataset [15]. Most of these efforts focus on concentration and overlook indirect dependencies in their analysis.

Other complementary efforts study internal backend infrastructure dependencies for debugging (e.g., [37]).

**Understanding CDN and hosting:** Several studies have focused on understanding CDNs and hosting infrastructure [2, 12, 34, 42, 56]. Other work maps the growing infrastructure and edge deployment by popular content providers (e.g., [8]). Recent work points out an increasing adoption of DDoS protection services by Websites [30]. Cangialosi et al.,[9] analyzes prevalence of private key sharing by websites with hosting providers. These are complementary to our work as we do not consider the hosting infrastructure and capacity bottlenecks of service providers.

**Internet measurement:** Zmap [19] and Censys [18] present mechanisms to scan the Internet to understand vulnerable services. Our focus on web infrastructure is complementary to this work. Other work has analyzed the use of TLS, the certificate ecosystem, and the use of Certificate Revocation in the wild (e.g., [13, 14, 32, 39, 63, 67]). These suggest potential attacks that could be executed via the third party services we analyze here.

**Website complexity and performance:** Butkiewicz et al., study the impact of the complexity of the website as measured in terms of the number of third-party objects on the website load time (e.g., [7]). Other work analyzes the critical paths to understand if and how specific content affects the page load time (e.g., [64]). However, our focus is on the infrastructure services at a higher level than individual websites. Other work focuses on the privacy implications of the tracking services that appear on the website (e.g., [33, 36, 51]). This is orthogonal to our work.

## 8 Discussion

We conclude with implications of our findings and some recommendations for different stakeholders.

### 8.1 Trends of concern

**Critical Inter-service Dependencies:** Critical dependencies between service providers (Section 6) further increase the websites' exposure to risk resulting from their critical dependencies. As a result of indirect dependencies, the number of critical dependencies per website increases, e.g., 25% of top-100K websites have 3 critical dependencies per website as compared to 9.6% when we just consider direct dependencies. Moreover, indirect dependencies amplify a provider's impact; e.g., CloudFlare impacts 44% of the top-100K websites vs. 24% when we only consider direct dependencies, DNSMadeEasy and Incapsula impact 25% of the websites from 1-2%.

**Lessons Learned?:** Between 2016 to 2020, we observe 1% to 5% increase in critical dependency in websites (Section 4). While the providers directly impacted by Dyn have adapted somewhat, it seems that lessons from the Dyn attack have been only acted upon by a handful who were directly impacted.

**Increasing Concentration:** As others also note, concentration is increasing (Figure 6). These potential single points of failure can become attractive targets for malicious actors.

**Prevalence across sectors:** Our preliminary case studies of two other sectors (Section 6) suggest that third-party dependencies proliferate across sectors leaving them vulnerable to Dyn-like incidents [24].

### 8.2 Recommendations

**Websites:** An obvious recommendation for websites is that they need to build in more resilience and redundancy when using third-party services. While many of these third-party providers have multiple points of presence, which introduces some redundancy within their infrastructure, they are not immune to failures e.g., the incidents mentioned in Section 2. We also acknowledge that third party providers have certain benefits such as better quality of service, higher capacity etc. which small websites cannot afford on their own. Moreover, websites need to understand the hidden dependencies of the third-party services they use as they maybe indirectly exposed to potential threats. For example, if a website uses multiple CDN providers but those CDNs use the same DNS provider. Similarly, if indirect dependencies of a website are also redundantly provisioned, it makes the website less prone to outages. The types of analysis we have performed can be made available as a neutral service that websites can query before making business decisions.

**Service Providers:** Service providers should support and encourage redundancy; e.g., Dyn offers "secondary" DNS configurations as a service [17]. Moreover, service providers should be more transparent about attacks they see and potential resiliency measures they have in place. Furthermore, they should also be judicious, and transparent, in their use of other third-party services as these transitive dependencies amplify the impact.

### 8.3 Future Work

A few natural future directions include incorporating analysis of dependencies among websites, measuring capacity of service providers to give a better picture of their individual vulnerability, designing a defense metric that utilizes these measurements to estimate robustness of a website. We also envision using our framework to build a service that given a website analyzes its complete dependency structure and enables the website administrator to make informed policy decisions on choosing new service providers. Moreover, this dependency analysis can also be extended to study service-level dependencies e.g., payment processors, messaging platforms, CRM etc. We can also perform interesting case studies on e-commerce, education, or government sector etc., to analyse their third party dependencies.. Moreover, to categorize third party providers, we can also extend our analysis to use other heuristics such as abuse emails etc. This can further increase the robustness of our algorithms mentioned in Section 3.

## Availability

Our code is also publically available [2]. All of the information we analyze is publicly visible and does not raise any ethical issues.

## Acknowledgments

We thank our shepherd, Kimberly C. Claffy, and all anonymous reviewers for their insightful comments on this paper. We would also like to acknowledge the contributions of Carolina Zarate and Hanrou Wang in helping with a similar dependency analysis of popular websites in 2016 and thank Antonis Manousis for his valuable feedback. This research has been supported in part by NSF awards TWC-1564009, SaTC-1801472, and CNS-1700521.

## References

[1] Abhishta Abhishta, Roland van Rijswijk-Deij, and Lambert JM Nieuwenhuis. 2019. Measuring the impact of a successful DDoS attack on the customer behaviour of managed DNS service providers. *ACM SIGCOMM Computer Communication Review* 48, 5 (2019), 70–76.

[2] Bernhard Ager, Wolfgang Mühlbauer, Georgios Smaragdakis, and Steve Uhlig. 2011. Web content cartography. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*. 585–600.

[3] Mark Allman. 2018. Comments on DNS robustness. In *Proceedings of the Internet Measurement Conference 2018*. 84–90.

[4] Samantha Bates, John Bowers, Shane Greenstein, Jordi Weinstock, Yunhan Xu, and Jonathan Zittrain. 2018. *Evidence of Decreasing Internet Entropy: The Lack of Redundancy in DNS Resolution by Major Websites and Services.* Technical Report. National Bureau of Economic Research.

[5] Kris Beevers. 2016. A Note From NS1's CEO: How We Responded To Last Week's Major, Multi-Faceted DDoS Attacks. https://ns1.com/blog/how-we-responded-to-last-weeks-major-multi-faceted-ddos-attacks.

[6] Hanno Bock. 2017. The Problem with OCSP Stapling and Must Staple and why Certificate Revocation is still broken, 2017. *URL https://blog. hboeck. de/archives/886-The-Problem-with-OCSP-Stapling-and-Must-Staple-and-why-Certificate-Revocation-is-still-broken. html* (2017).

[7] Michael Butkiewicz, Harsha V Madhyastha, and Vyas Sekar. 2011. Understanding website complexity: measurements, metrics, and implications. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*. 313–328.

[8] Matt Calder, Xun Fan, Zi Hu, Ethan Katz-Bassett, John Heidemann, and Ramesh Govindan. 2013. Mapping the expansion of Google's serving infrastructure. In *Proceedings of the 2013 conference on Internet measurement conference*. 313–326.

[9] Frank Cangialosi, Taejoong Chung, David Choffnes, Dave Levin, Bruce M Maggs, Alan Mislove, and Christo Wilson. 2016. Measurement and analysis of private key sharing in the https ecosystem. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. 628–640.

[10] CDNFinder. 2020. Webapp and cli-tool to detect CDN usage of websites. https://github.com/turbobytes/cdnfinder. Accessed: May 23, 2020.

[11] Ericka Chickowski. April 7, 2020. Cyberattacks Against Pandemic-Stressed Healthcare Organizations. https://securityboulevard.com/2020/04/7-cyberattacks-against-pandemic-stressed-healthcare-orgs/. Accessed: April 23, 2020.

[12] David Choffnes, Jilong Wang, et al. 2017. CDNs meet CN an empirical study of CDN deployments in China. *IEEE Access* 5 (2017), 5292–5305.

[13] Taejoong Chung, Yabing Liu, David Choffnes, Dave Levin, Bruce MacDowell Maggs, Alan Mislove, and Christo Wilson. 2016. Measuring and applying invalid SSL certificates: The silent majority. In *Proceedings of the 2016 Internet Measurement Conference*. 527–541.

[14] Taejoong Chung, Jay Lok, Balakrishnan Chandrasekaran, David Choffnes, Dave Levin, Bruce M Maggs, Alan Mislove, John Rula, Nick Sullivan, and Christo Wilson. 2018. Is the Web Ready for OCSP Must-Staple?. In *Proceedings of the Internet Measurement Conference 2018*. 105–118.

[15] Matteo Dell'Amico, Leyla Bilge, Ashwin Kayyoor, Petros Efstathopoulos, and Pierre-Antoine Vervier. 2017. Lean on me: Mining internet service dependencies from large-scale dns data. In *Proceedings of the 33rd Annual Computer Security Applications Conference*. 449–460.

[16] Digicert. 2020. DigiCert Completes Acquisition of Symantec's Website Security and Related PKI Solutions. https://www.digicert.com/news/digicert-completes-acquisition-of-symantec-ssl/.

[17] Dyn Secondary DNS. May 23, 2020. Dyn Secondary DNS Information. https://help.dyn.com/standard-dns/dyn-secondary-dns-information/.

[18] Zakir Durumeric, David Adrian, Ariana Mirian, Michael Bailey, and J Alex Halderman. 2015. A search engine backed by Internet-wide scanning. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. 542–553.

[19] Zakir Durumeric, Eric Wustrow, and J Alex Halderman. 2013. ZMap: Fast Internet-wide scanning and its security applications. In *Presented as part of the 22nd {USENIX} Security Symposium ({USENIX} Security 13)*. 605–620.

[20] Fastly. October 21, 2016. Fastly outage. https://www.fastly.com/security-advisories/widespread-dyn-dns-outage-affecting-fastly-customers. Accessed: May 23, 2020.

[21] GlobalSign October 13, 2016. Globalsign certificate revocation issue. https://www.globalsign.com/en/status. Accessed: May 23, 2020.

[22] Ariya Hidayat et al. 2013. PhantomJS. *Computer software. PhantomJS. Vers* 1, 7 (2013).

[23] Simon Hildrew and Jenny Sivapalan. 2016. Multiple DNS: synchronising Dyn to AWS Route 53. https://www.theguardian.com/info/developer-blog/2016/dec/23/multiple-dns-synchronising-dyn-to-aws-route-53.

[24] Scott Hilton. Oct 26, 2016. Dyn analysis summary of friday october 21 attack. http://dyn.com/blog/dyn-analysis-summary-of-friday-october-21-attack/. Accessed: May 23, 2020.

[25] Rand Hindi. February 28, 2017. Thanks for breaking our connected homes, Amazon. https://medium.com/snips-ai/thanks-for-breaking-our-connected-homes-amazon-c820a8849021.

[26] Nguyen Phong Hoang, Arian Akhavan Niaki, Michalis Polychronakis, and Phillipa Gill. 2020. The web is still small after more than a decade. *ACM SIGCOMM Computer Communication Review* 50, 2 (2020), 24–31.

[27] Hsu-Chun Hsiao, Tiffany Hyun-Jin Kim, Yu-Ming Ku, Chun-Ming Chang, Hung-Fang Chen, Yu-Jen Chen, Chun-Wen Wang, and Wei Jeng. 2019. An Investigation of Cyber Autonomy on Government Websites. In *The World Wide Web Conference*. 2814–2821.

[28] IETF. Mar 4, 2018. Consolidation. https://www.ietf.org/blog/consolidation/.

[29] Muhammad Ikram, Rahat Masood, Gareth Tyson, Mohamed Ali Kaafar, Noha Loizon, and Roya Ensafi. 2019. The chain of implicit trust: An analysis of the web third-party resources loading. In *The World Wide Web Conference*. 2851–2857.

[30] Mattijs Jonker, Anna Sperotto, Roland van Rijswijk-Deij, Ramin Sadre, and Aiko Pras. 2016. Measuring the adoption of DDoS protection services. In *Proceedings of the 2016 Internet Measurement Conference*. 279–285.

[31] Peter Koch. 1999. Recommendations for DNS SOA Values. (1999).

[32] Platon Kotzias, Abbas Razaghpanah, Johanna Amann, Kenneth G Paterson, Narseo Vallina-Rodriguez, and Juan Caballero. 2018. Coming of age: A longitudinal study of tls deployment. In *Proceedings of the Internet Measurement Conference 2018*. 415–428.

[33] Balachander Krishnamurthy and Craig Wills. 2009. Privacy diffusion on the web: a longitudinal perspective. In *Proceedings of the 18th international conference on World wide web*. 541–550.

[34] Balachander Krishnamurthy, Craig Wills, and Yin Zhang. 2001. On the use and performance of content distribution networks. In *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*. 169–182.

[35] Deepak Kumar, Zane Ma, Zakir Durumeric, Ariana Mirian, Joshua Mason, J Alex Halderman, and Michael Bailey. 2017. Security challenges in an increasingly tangled web. In *Proceedings of the 26th International Conference on World Wide Web*. 677–684.

[36] Adam Lerner, Anna Kornfeld Simpson, Tadayoshi Kohno, and Franziska Roesner. 2016. Internet jones and the raiders of the lost trackers: An archaeological study of web tracking from 1996 to 2016. In *25th {USENIX} Security Symposium*

[2] https://github.com/AqsaKashaf/Analyzing-Third-party-Dependencies.git

*({USENIX} Security 16)*.

[37] Zhichun Li, Ming Zhang, Zhaosheng Zhu, Yan Chen, Albert G Greenberg, and Yi-Min Wang. 2010. WebProphet: Automating Performance Prediction for Web Services.. In *NSDI*, Vol. 10. 143–158.

[38] Public Suffix List. [n.d.]. Mozilla Public Suffix List.

[39] Yabing Liu, Will Tome, Liang Zhang, David Choffnes, Dave Levin, Bruce Maggs, Alan Mislove, Aaron Schulman, and Christo Wilson. 2015. An end-to-end measurement of certificate revocation in the web's PKI. In *Proceedings of the 2015 Internet Measurement Conference*. 183–196.

[40] Shelby Livingston. October 21, 2016. Athenahealth, Allscripts websites down amid nationwide hack. https://www.modernhealthcare.com/article/20161021/NEWS/161029973/athenahealth-allscripts-websites-down-amid-nationwide-hack.

[41] Chaoyi Lu, Baojun Liu, Zhou Li, Shuang Hao, Haixin Duan, Mingming Zhang, Chunying Leng, Ying Liu, Zaifeng Zhang, and Jianping Wu. 2019. An End-to-End, Large-Scale Measurement of DNS-over-Encryption: How Far Have We Come?. In *Proceedings of the Internet Measurement Conference*. 22–35.

[42] Srdjan Matic, Gareth Tyson, and Gianluca Stringhini. 2019. Pythia: a Framework for the Automated Analysis of Web Hosting Environments. In *The World Wide Web Conference*. 3072–3078.

[43] Tobias Mueller, Daniel Klotzsche, Dominik Herrmann, and Hannes Federrath. 2019. Dangers and Prevalence of Unprotected Web Fonts. In *2019 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*. IEEE, 1–5.

[44] Multi-CDN. 2020. Multi-CDN Strategies. https://ns1.com/multi-cdn.

[45] Arun Natarajan, Peng Ning, Yao Liu, Sushil Jajodia, and Steve E Hutchinson. 2012. *NSDMiner: Automated discovery of network service dependencies*. IEEE.

[46] Newsweek. 2020. Top Hospitals in the U.S. https://www.newsweek.com/best-hospitals-2020/united-states. Accessed: May 23, 2020.

[47] Nick Nikiforakis, Luca Invernizzi, Alexandros Kapravelos, Steven Van Acker, Wouter Joosen, Christopher Kruegel, Frank Piessens, and Giovanni Vigna. 2012. You are what you include: large-scale evaluation of remote javascript inclusions. In *Proceedings of the 2012 ACM conference on Computer and communications security*. 736–747.

[48] Karlis Podins and Arturs Lavrenovs. 2018. Security Implications of Using Third-Party Resources in the World Wide Web. In *2018 IEEE 6th Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE)*. IEEE, 1–6.

[49] Alexa Traffic Rank. 2020. List of most popular web sites.

[50] Dark Reading. October 24, 2019. Eight-Hour DDoS Attack Struck AWS Customers. https://www.darkreading.com/cloud/eight-hour-ddos-attack-struck-aws-customers/d/d-id/1336165. Accessed: May 20, 2020.

[51] Franziska Roesner, Tadayoshi Kohno, and David Wetherall. 2012. Detecting and defending against third-party tracking on the web. In *Presented as part of the 9th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 12)*. 155–168.

[52] Jukka Ruohonen, Joonas Salovaara, and Ville Leppänen. 2018. Crossing cross-domain paths in the current web. In *2018 16th Annual Conference on Privacy, Security and Trust (PST)*. IEEE, 1–5.

[53] Quirin Scheitle, Oliver Hohlfeld, Julien Gamba, Jonas Jelten, Torsten Zimmermann, Stephen D Strowes, and Narseo Vallina-Rodriguez. 2018. A long way to the top: Significance, structure, and stability of internet top lists. In *Proceedings of the Internet Measurement Conference 2018*. 478–493.

[54] Catherine Shu. February 24, 2020. Petnet's smart pet feeder system is back after a week-long outage. https://techcrunch.com/2020/02/24/petnets-smart-pet-feeder-system-is-back-after-a-week-long-outage-but-customers-are-still-waiting-for-answers/.

[55] Milivoj Simeonovski, Giancarlo Pellegrino, Christian Rossow, and Michael Backes. 2017. Who controls the internet? analyzing global threats using property graph traversals. In *Proceedings of the 26th International Conference on World Wide Web*. 647–656.

[56] Rachee Singh, Arun Dunna, and Phillipa Gill. 2018. Characterizing the deployment and performance of multi-cdns. In *Proceedings of the Internet Measurement Conference 2018*. 168–174.

[57] Internet Society. February 26, 2019. Consolidation in the Internet Economy. https://www.internetsociety.org/news/press-releases/2019/internet-society-launches-research-project-to-understand-the-effects-of-consolidation-in-the-internet-economy/.

[58] Akamai Multi-CDN Support. 2020. Akamai - We offer support for multiple CDNs. https://learn.akamai.com/en-us/webhelp/media-acceleration/media-acceleration-sdk-integration-guide-for-javascript/GUID-E246743C-703D-4885-B934-171788539187.html.

[59] Fastly Multi-CDN Support. 2020. Fastly Launches Cloud Optimizer to Boost Observability and Control in Multi-Cloud and Multi-CDN Infrastructures. https://www.fastly.com/press/press-releases/fastly-launches-cloud-optimizer-boost-observability-and-control-multi-cloud-and-multi-cdn-infrastructures.

[60] Samaneh Tajalizadehkhoob, Maciej Korczyński, Arman Noroozian, Carlos Ganán, and Michel van Eeten. 2016. Apples, oranges and hosting providers: Heterogeneity and security in the hosting market. In *NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 289–297.

[61] Kevin C. Tofel. February 26, 2020. It's time for smart home devices to have local failover options during cloud outages. https://staceyoniot.com/smart-home-devices-cloud-outage-vs-local/.

[62] Tobias Urban, Martin Degeling, Thorsten Holz, and Norbert Pohlmann. 2020. Beyond the front page: Measuring third party dynamics in the field. In *Proceedings of The Web Conference 2020*. 1275–1286.

[63] Benjamin VanderSloot, Johanna Amann, Matthew Bernhard, Zakir Durumeric, Michael Bailey, and J Alex Halderman. 2016. Towards a complete view of the certificate ecosystem. In *Proceedings of the 2016 Internet Measurement Conference*. 543–549.

[64] Xiao Sophia Wang, Aruna Balasubramanian, Arvind Krishnamurthy, and David Wetherall. 2013. Demystifying page load performance with WProf. In *Presented as part of the 10th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 13)*. 473–485.

[65] webXray. June 29, 2018. webXray Domain Owner List. https://github.com/timlib/webXray_Domain_Owner_List.

[66] Eric A Young, Tim J Hudson, and R Engelschall. 2011. Openssl: The open source toolkit for ssl/tls.

[67] Liang Zhu, Johanna Amann, and John Heidemann. 2016. Measuring the latency and pervasiveness of TLS certificate revocation. In *International Conference on Passive and Active Network Measurement*. Springer, 16–29.