Security Challenges in an Increasingly Tangled Web

Deepak Kumar[†] Zane Ma[†] Zakir Durumeric[‡†] Ariana Mirian[‡]
Joshua Mason[†] J. Alex Halderman[‡] Michael Bailey[†]
[†]University of Illinois, Urbana Champaign [‡]University of Michigan
{dkumar11, zanema2, joshm, mdbailey}@illinois.edu {zakir, amirian, jhalderm}@umich.edu

ABSTRACT

Over the past 20 years, websites have grown increasingly complex and interconnected. In 2016, only a negligible number of sites are dependency free, and over 90% of sites rely on external content. In this paper, we investigate the current state of web dependencies and explore two security challenges associated with the increasing reliance on external services: (1) the expanded attack surface associated with serving unknown, implicitly trusted third-party content, and (2) how the increased set of external dependencies impacts HTTPS adoption. We hope that by shedding light on these issues, we can encourage developers to consider the security risks associated with serving third-party content and prompt service providers to more widely deploy HTTPS.

Keywords

website complexity, HTTPS adoption, privacy/tracking

1. INTRODUCTION

Since its inception in 1989, the Internet community has had to cope with the decision to allow HTTP pages to load third-party content on the World Wide Web. From the need to limit the access of malicious websites through the same-origin policy, to coping with long page load times through fine-grained resource scheduling, the dependency graph that underlies websites has been of critical importance in understanding the security and performance of the web. Existing measurements of this tangle of dependencies predate an explosion of advertising and tracking technologies and an increased reliance on shared platforms.

In order to untangle the interdependent web, we extend the headless version of Google Chromium to determine the services and networks that the Alexa Top Million sites load resources from. We find that sites load a median of 73 resources, 23 of which are loaded from external domains — each twice the corresponding figure from five years prior [5]. We show that a small number of third parties serve resources for a large fraction of sites, with Google, Facebook, and Twitter appearing on 82%, 34%, and 11% of the top million sites respectively. Investigating the top networks that provide these remote resources, we find that shared platforms deliver content to many sites — content distribution networks (CDNs) serve content to 60% of the top million and cloud providers serve resources to 37% of the top million.

The increase in the number of externally loaded resources and their distribution yield an attractive attack vector for compromising large numbers of clients through shared dependencies and platforms in use across popular sites. Perhaps more troubling, we find that 33% of the top million sites load unknown content *indirectly* through at least one third-party, exposing users to resources that site operators have no relationship with. Coupled with the observation that 87% of sites execute active content (e.g., JavaScript) from an external domain, a tangled attack landscape emerges. We note that modern resource integrity techniques, such as subresource integrity (SRI), are not applicable to unknown content, and it remains an open problem to mitigate these kinds of attacks. The complex interdependencies on the web have consequences beyond security. We consider, as one example, the widespread deployment of HTTPS. We find that 28% of HTTP sites are blocked from upgrading to HTTPS by active-content dependencies that are not currently available over HTTPS. This only accounts for active content — 55% of sites loaded over HTTP rely on an active or passive HTTP resource that is not currently available over HTTPS. Encouragingly, we find that 45% of HTTP sites could migrate to full HTTPS by changing the protocol in the resources they load. However, the community still has much work to do to encourage the widespread adoption of HTTPS.

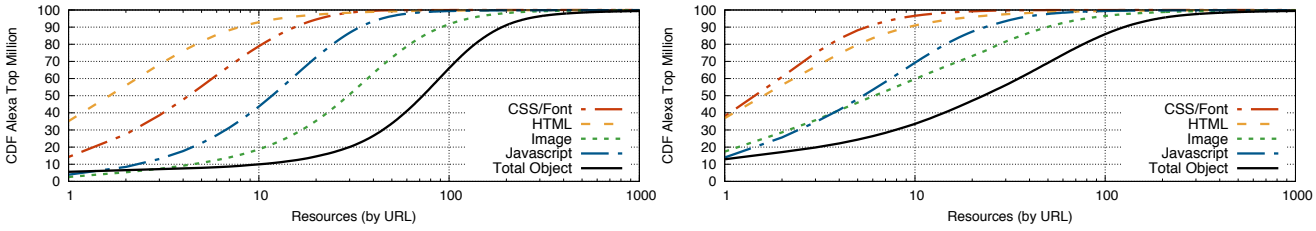
We hope that by bringing to light these security and privacy challenges introduced by an increasingly tangled web, we remind the community of the many challenges involved with securing this distributed ecosystem. To facilitate further research on the topic, we are releasing our headless browser as an open source project that is documented and packaged for measuring the composition of sites.

2. DATA COLLECTION

Our analysis focuses on the resources (e.g., images, scripts, style sheets, etc.) that popular websites load when rendered on a desktop browser. To track these dependencies, we extended the headless version of Google Chromium [12] to record websites' network requests, which we use to reconstruct the resource tree for each site. Unlike prior web resource studies, which have used a bipartite graph model of web pages [5, 13, 25], we build a tree that preserves the relationships between nested dependencies.

We visited all of the sites in the Alexa Top Million domains [1] on October 5–7, 2016 from the University of Michigan using our headless browser. We specifically attempted to load the root page of each site over HTTP (e.g., http://google.com). If we could not resolve the root domain, we instead attempted to fetch www.domain.com. We stopped loading additional content 10 seconds after the last resource, or after 30 seconds elapsed. Our crawl took approximately 48 hours

WWW 2017, April 3–7, 2017, Perth, Australia.
ACM 978-1-4503-4913-0/17/04.
http://dx.doi.org/10.1145/3038912.3052686



(a) **All Resources** (b) **External Resources**

Figure 1: **CDF of Resources** — We show the CDF of the number of resources (by URL) loaded by the top million websites.

on 12 entry-level servers. We were able to successfully load the root page for 944,000 sites. 15K domains did not resolve, 13K timed out, 24K provided an HTTP error (i.e., non-200 response), and 5K could not be rendered by Chromium.

After completing the crawl, we annotated each resource with additional metadata. We first made an additional HTTP and HTTPS request for each resource using ZGrab [7, 8] to determine whether the resource was available over HTTPS. We also captured the SHA-1 hash of each resource, the AS it was loaded from, and the Maxmind [21] geolocation of the web server. We are releasing our headless browser at https://github.com/zmap/zbrowse and our dataset at https://scans.io/study/tangled.

Ethical Considerations As with any active scanning methodology, there are many ethical considerations at play. In our crawl, we only loaded sites in the Alexa Top Million, for which our traffic should be negligible — at most equivalent to a user loading a page three times. The only exception to this were sites for which we measured temporal changes, which we loaded hourly over a five day period. We followed the best practices defined by Durumeric et al. [8] and refer to their work for more detailed discussion of the ethics of active scanning.

Determining Content Origin Throughout our analysis, we differentiate between local and external resources. We make this distinction using the Public Suffix List [23]. For example, we would not consider staticxx.facebook.com to be hosted on a different domain than www.facebook.com, but we do consider gstatic.com to be external from google.com. In order to group resources by entity rather than domain, we additionally label each resource with the AS it was loaded from.

Root vs. Internal Pages Our study approximates the resources that websites load by analyzing the dependencies loaded by each website's root page. To determine whether this is representative of the entire website, we analyzed five internal pages for a random 10K sites in our dataset.[1] On average, the root page of each site loads content from 87% of the union of external domains that all pages depend on and 86% of the external ASes. This is nearly double that of just the internal pages, which contribute 40% of the total domains and 44% of the total networks. In other words, while the root page of a site does not typically load content from all of a site's dependencies, it is representative of most of the dependencies on a site as a whole.

Temporal Changes We also measured how dependencies change over time by loading a random 1,000 domains every hour for five days and tracking the new content introduced in each crawl. We find that 57% of external domains are captured in the first crawl, 5.7% in the second, and less than 1% in the third. A single snapshot does not capture all of the dependencies for a site, but the additional data collected from subsequent runs quickly diminishes.

[1] We identified five pages for each site using Bing.

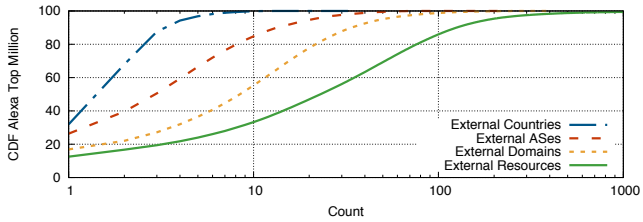


Figure 2: **External Dependencies** — Websites load a median 23 external resources from 9 external domains, 3 external ASes, and 1 foreign country.

Figure 3: **LA Times** — The LA Times homepage loads 540 resources from 267 unique IP addresses in 58 ASes and 8 countries.

3. WEB COMPLEXITY

There has been a stark increase in website complexity over the past five years. In 2011, Butkiewicz et al. found that the 20,000 most popular websites loaded a median 40 resources and 6 JavaScript resources [5]. Today, the top 20K websites load twice the number of scripts and total resources. A negligible number of sites with user content are altogether dependency free, and the top million sites now load a median 73 resources. Sites are not only including a larger number of resources, but they are also increasingly relying on external websites and third-party networks. In 2011, 30% of resources were loaded from an external domain. In the last five years, this has nearly doubled, and, today, the majority (64%) of resources are external. More than 90% of sites have an external dependency and 89% load resources from an external network (Figure 2).

3.1 Site Composition

As shown in Figure 1a, there is a large variation in the number of resources that sites load. The top million sites load a median 73 resources, but popular, media-heavy sites tend to include many more. News and sports sites have the most dependencies and load a median 247 and 207 resources, respectively. News and sports also

Resource Type	All	External
Images	58.8%	51.8%
JavaScript	21.6%	25.5%
Data (HTML, XML, JSON, text)	9.4%	14.3%
CSS	7.9%	4%
Other	2.3%	4.4%

Figure 4: **Types of Resource Loaded by the Top Million Sites**—We show the breakdown of types of resources loaded by the top million sites. External content on sites skews more towards scripts and data when compared to all resources.

Domain	Top 1M	Domain	Top 1M
google-analytics.com	67.8%	ajax.googleapis.com	23.1%
gstatic.com	50.1%	googlesyndication.com	19.6%
fonts.googleapis.com	42.8%	googleadservices.com	14.1%
doubleclick.net	40.5%	twitter.com	12.8%
facebook.com	33.7%	fbcdn.net	10.7%
google.com	33.2%	adnxs.com	10.5%
facebook.net	27.4%		

Figure 5: **Domains Loaded by ≥10% of Sites**—The domains that are commonly included on at least 10% of sites are owned by just four companies, Google, Facebook, Twitter, and AppNexus. Google controls 8 of the top 10 most loaded domains.

load the most content from external services: 80% of their resources are external. In one of the more complex cases, the LA Times homepage includes 540 resources from nearly 270 IP addresses, 58 ASes, and 8 countries (Figure 3). CNN—the most popular mainstream news site by Alexa rank—loads 361 resources.

Approximately 6% of sites do not have any dependencies. When we manually investigated these sites, we found that the vast majority do not use their root pages to serve user content. Instead, the domains are used to host media content at specific paths. The root page of mostazaweb.com.ar (an Argentinian fast food chain) loaded the most resources—nearly 20K. In general, sites with an extreme number of dependencies were either broken—loading resources an infinite number of times until timing out—or are less popular sites that load a large number of images.

Most resources are images (59%) and scripts (22%).[2] Of the 73 median files that sites load, 30 are images, 12 are scripts, and 5 are style sheets (Figure 4). Nearly all of the most commonly included resources (by file hash) support advertising/tracking programs and are served from external sites. The single most common file is a 1×1 white gif, which is used by an array of analytics and advertisement providers. More than 70% of sites include the pixel from an external domain and 91% of those load it from Google (e.g., as part of Google Analytics). The most common file unrelated to tracking or advertising is `jquery.migrate.min.js`, a WordPress dependency. It is the 11th most common file and is loaded on 13% of sites.

3.2 External Dependencies

Just over 90% of the top million sites have external dependencies, and more than two thirds of *all* resources are loaded from external sites. Despite the large number of external dependencies, there are only four companies—Google, Facebook, Twitter, and AppNexus—that serve content on more than 10% of the top million sites (Figure 5). The most frequently included external domain is google-analytics.com, which is present on 68% of websites. Sites

[2]We categorized resource types by analyzing the Content-Type HTTP header (e.g., *image/png*) and file extension, which allowed us to classify 99.8% of dependencies.

Category	% of Ext. Resources	Top 1M Domains	
Analytics/Tracking	8.2%	731,056	(75.4%)
CDN/Static Content	8.4%	631,718	(65.2%)
Service/API	8.8%	377,363	(39.0%)
Advertising	13.9%	409,405	(42.2%)
Social Media	11.0%	385,103	(39.7%)
Unknown	50.3%	–	–

Figure 6: **External Dependency Purposes**—We categorized common dependencies that appear on more than 1% of sites. Analytics and tracking resources are the most commonly included category for Top Million websites, but advertising and social media dependencies account for more total resources loads.

Company	Type	Top 1M	Company	Type	Top 1M
Google	All	82.2%	MaxCDN	CDN	19.0%
Facebook	Social	34.1%	Edgecast	CDN	17.9%
Amazon EC2	Cloud	32.6%	Fastly	CDN	15.5%
Cloudflare	CDN	30.7%	SoftLayer	Cloud	11.8%
Akamai	CDN	20.3%	Twitter	Social	11.2%

Figure 7: **ASes that Serve Content for ≥10% of Sites**—Google resources are loaded on over 4 out of 5 sites in the top million sites. Other social, cloud, and CDN providers are significantly less prominent and serve content for 11–34% of sites.

also include an array of other Google services beyond their analytics program: 47% use AdWords/DoubleClick, 44% serve Google Fonts, and 43% call other Google APIs. Facebook and Twitter both provide social media plugins; AppNexus is a popular ad provider.

While there are only a handful of providers that serve content on more than 10% of sites, there is a long tail of dependencies that appear on a smaller number of domains: 22 domains are loaded by 5–10% of sites and 185 domains are loaded by 1–5%. We manually categorized the domains that more than 1% of sites depend on and find that most common dependencies are part of analytics/-tracking (29.4%) or advertising (29.0%) programs. We provide a detailed breakdown in Figure 6. We note that while analytics and tracking services are used by more websites than any other type of service, advertising accounts for the largest number of external dependencies.

3.3 Network Providers

While aggregating external resources by domain provides one perspective, this analysis fails to identify many of the lower layer services that are shared between websites. Cloud providers, CDNs, and network services have visibility into much of the same data as the websites themselves do, and their compromise could cascade onto the upstream services that rely on them. To measure the service providers that popular sites rely on, we aggregated resources by the AS they are served from.

At least 20% of sites depend on content loaded from Google, Facebook, Amazon, Cloudflare, and Akamai ASes. Five of the top ten networks most relied on are CDNs, two are cloud providers, and one (Google) serves several roles (Figure 7). The five largest CDNs serve 12% of all resources and content for 60% of the top million. Cloudflare and Akamai both have a large set of customers and serve a variety of files to many domains. Several lesser known CDNs are used by a surprisingly large number of sites. 10% of sites depend on MaxCDN to provide `bootstrap.min.js` and `font-awesome.min.css`. Fastly serves content for three sites that provide popular embedded content: imgur.com, shopify.com, and vimeo.com.

The two cloud providers that serve content to the largest number of sites in the top million, Amazon EC2 and Softlayer, serve 7.2% of all resources and are relied on by 37.3% of the top million sites. Though both cloud providers serve a varied set of customers, most domains load resources from these providers because several popular advertising services use them for hosting. Nine of the top ten domains served out of EC2 and SoftLayer belong to advertising campaigns; the only non-advertising related domain is s3.amazonaws.com — Amazon's storage platform.

While loading passive content (e.g., an image) from a CDN may not be a serious security concern for many websites, nearly 50% of the top million sites include JavaScript resources from a CDN, which poses a larger security risk. Large CDNs are typically well managed, however, there have historically been attacks against CDNs and they should be considered as part of a site's vulnerability profile. In 2013, MaxCDN — the official CDN for Bootstrap — was compromised by a rogue contractor [20]. The CDN served malicious JavaScript that exploited a Java vulnerability on tens of thousands of sites until operators could investigate and correct the problem — days later.

In light of such attacks, browsers introduced support for Subresource Integrity (SRI) [34], an HTML extension that allows developers to specify the expected cryptographic hash of a resource. SRI prevents a compromised service provider from modifying the expected content. Unfortunately, less than 1% of sites in our study use SRI, which indicates that CDNs are unnecessarily increasing the attack service for most popular sites.

4. IMPLICIT TRUST

Relying on a larger number of resources does not inherently pose a security risk. However, the expanding trust on external sites *does* represents an increase in the attack surface for both websites and end users. In many cases, we find that websites operators no longer know who they are trusting because external services load *implicitly trusted* content from third parties that are unknown to the main site operator.

To understand who sites are implicitly trusting, we analyzed the depth at which different resources are loaded. We denote resources that the main site directly includes as *explicitly trusted*, and objects loaded from third parties by external services as *implicitly trusted*. For example, if the New York Times loads advertising JavaScript from DoubleClick and DoubleClick loads additional content from a third-party ad provider such as SmartAdServer, we should state that the New York Times explicitly trusts DoubleClick, and implicitly trusts SmartAdServer. We note that the distinction between implicit trust is dependent not only on depth, but also owner. If a DoubleClick script loaded additional content from its own servers or the original site, we would not mark this as implicitly trusted.

Nearly 33% of the top million sites include at least one implicit resource, and 20% of *all* external resources are loaded implicitly for the top 100K websites. Websites most commonly implicitly trust images (48%). In this situation, there is a modest security risk because the ad provider could serve a deceiving ad (e.g., phishing content). More worryingly, 32% of implicitly trusted resources are scripts and 3% are style sheets. An astounding 9% of the top million and 13% of the top 100K sites load implicitly trusted scripts from third parties.

One of the primary reasons this occurs is because real-time ad bidding has become a standard practice and ad exchanges are commonly serving content directly from bidders [37]. DoubleClick loads the most implicit content and is responsible for implicitly trusted resource on 9.6% of the top million sites (Figure 8). We note that while Facebook, Google, and YouTube also appear near the top of the list, they are primarily loading additional content from

Includes Implicit	% Top 1M	Implicit Domain	% Top 1M
doubleclick.net	9.6%	fbcdn.net	9.2%
facebook.com	9.3%	gstatic.com	8.9%
google.com	4.7%	adsrvr.org	2.9%
youtube.com	3.3%	agkn.com	2.8%
adlegend.com	2.0%	adnxs.com	2.3%
casalemedia.com	1.4%	s3.amazonaws.com	2.1%
sharethis.com	1.3%	jwplatform.com	2.1%
googlesyndication.com	1.1%	cloudflare.com	2.0%
vk.com	1.0%	media6degrees.com	1.9%
2mdn.net	0.9%	adsymptotic.com	1.8%

Figure 8: **Sources of Implicitly Trusted Content** — We show the domains that load the most implicitly-trusted content and the domains that are most implicitly-trusted.

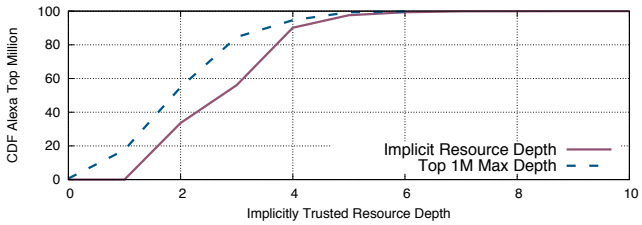


Figure 9: **Implicit Depth CDF** — We present a CDF of the depths at which implicitly trusted resources are loaded for the top million sites. The median depth for implicitly trusted resources is 3, indicating that even explicitly trusted services have limited visibility into the content loaded on the parent websites.

fbcdn.com and gstatic.com, and do not pose the same security challenges because they are operated by the same organization. However, beyond these domains, we find a set of lesser known ad providers. For example, Casale Media loads implicit content from 65 third party domains. The top three most popular third-party domains are only loaded 5–7% of the time, which limits the investigation that an operator could do into the third-party operators that they are implicitly trusting.

News and sports sites trust the most implicit content, primarily due to their heavy reliance on advertising. 70% of the top news and sports sites implicitly trust at least one resource. We noted a similar trend in the total number of external resources loaded earlier for these categories, indicating that news and sports sites have a markedly larger attack surface than other types of sites. Continuing our previous example with the LA times, we find that their homepage loads 64 implicitly trusted resources from 44 external domains. The Huffington Post loads 66 implicitly trusted resources from 51 unique, implicitly-trusted domains.

Intrinsic to implicit trust is the depth at which the resource is loaded. In this context, we define the depth of a resource as the number of nested resource loads from the root page. For example, an image included by a page would have depth 1. As resources load, we capture the depth at which they are loaded from, with the root page at depth 0. As shown in Figure 9, implicitly trusted resources are loaded at a median depth of 3, indicating that most implicitly trusted resources are loaded with at least two levels of indirection. This indirection further decreases the visibility that the site operator has into the resources the site loads.

The inclusion of implicitly trusted resources greatly increases the attack surface for websites, as an attacker only needs to compromise one of a large number of ad providers rather than the main site itself. This attack is not merely theoretical. In 2016, several mainstream websites, including the New York Times, BBC, Newsweek, NFL,

URL	Owner	T10K	T100K	T1M
b.scorecardresearch.com	comScore	27.2%	12.4%	5.3%
*.casalemedia.com	Casale Media	22.1%	10.7%	2.5%
*.baidu.com	Baidu	7.8%	7.9%	1.7%
*.sharethis.com	ShareThis	2.1%	2.6%	1.6%
www.statcounter.com	StatCounter	1.2%	1.3%	1.5%
cdn.turn.com	Turn Inc.	7.8%	4.0%	1.3%
cloudfront-labs.amazonaws.com	Alexa	11.0%	4.1%	1.2%
global.ib-ibi.com	Network Sol.	5.3%	3.1%	1.2%
a.adroll.com	AdRoll	1.9%	1.8%	1.1%
admaym.com	WideOrbit	3.5%	2.0%	0.8%
cdn.rubiconproject.com	Rubicon	6.8%	2.2%	0.6%

Figure 10: **Most Common HTTPS Blocking Domains**—We show the top sites that block the deployment of HTTPS and their prevalence in the top million sites. A site blocks the deployment of HTTPS if hosts a resource that appears on a site over HTTP and is not yet available over HTTPS.

AOL, MSN, and The Weather Channel began serving ransomware to customers after attackers acquired an expired domain associated with a trusted ad provider [19]. In each case, the main site did not explicitly trust the compromised domain, but rather included ads from one of several trusted ad providers: Google, AppNexus, AOL, and Rubicon, all of whom were affected.

Unfortunately, site owners have no mechanism for verifying the implicit content that they serve ahead of time, and the security community has yet to develop mechanisms for easily constraining the JavaScript that ads execute. As we will discuss in the next section, this tangle of resource dependencies complicates site management and may also be preventing parts of the web from migrating to HTTPS.

5. HTTPS BLOCKERS

There is a concerted effort to move the web to HTTPS. Standards bodies, including the IETF, IAB, and W3C, have called for securing the web by means of HTTPS [11,14,30,33]. Mozilla has announced plans to deprecate insecure HTTP connections, including slowly phasing out features for existing HTTP sites and setting a date after which new features will only be enabled for HTTPS sites [3]. Google will start marking HTTP sites as insecure in the future [26]. Despite these initiatives, only 35% of the top million websites support browser-trusted HTTPS. Even fewer serve their primary web traffic over HTTPS. Nearly 20% of the sites that support HTTPS immediately redirect users back to HTTP. Only 46% of HTTPS sites redirect users from HTTP to HTTPS, and even fewer (11%) include an HSTS header to indicate that future connections should be strictly served over HTTPS.

There have been numerous reports that service providers are blocking websites from upgrading to HTTPS. Wired reported in April 2016 that: "The biggest challenge, however, was advertising. While some ad networks and exchanges support HTTPS, adoption is spotty. This means that publishers that rush to switch to HTTPS risk not being able to work with certain ad partners, which in turn means potentially losing out on revenue" [32]. The Washington Post made a similar claim: "Ask any developer at a major media organization what the biggest hurdle to HTTPS adoption is, and the answer is always going to be advertising. [...] For the ads that come from ad exchanges, what's inside is even more of a black box—we have no idea what resources they will include, and no way of preventing specific resources." [36]. The New York Times stated that "If the assets for an advertisement aren't able to serve over an HTTPS channel, the advertisement will probably not display on the page, directly affecting revenue. It can be difficult to determine if each advertisement will load over HTTPS. Considering the importance of advertisements, this is very likely to be a significant hurdle to many media organizations' implementation of HTTPS." [15]. In this section, we investigate these claims and document the services that are preventing websites from upgrading to HTTPS.

In order for a site to migrate to HTTPS, all active content (e.g., JavaScript or CSS) on the site must be loaded over HTTPS. This is because browsers will not execute active content that is delivered over HTTP for an HTTPS page. To understand whether sites are blocked from migrating, we examined what external, active content is loaded over HTTP and whether this content is yet available over HTTPS. We find that 56.5% of HTTP-only sites in the top million load active content over HTTP from at least one external domain. Fortunately, 40% of the resources requested over HTTP are already available over HTTPS, and 45% of HTTP sites can immediately migrate to HTTPS by updating the protocol in their resource URLs. Unfortunately, 28% of HTTP-only sites load *active* content from external domains that do not support HTTPS. In other words, 28% of HTTP-only sites are blocked from migrating to HTTPS by a resource they depend on. While only active content blocks a site from migrating to HTTPS, sites will not be fully served over HTTPS until all content is served over HTTPS. We find that 55% of the top HTTP-only sites depend on external content that is not yet accessible over HTTPS, and thus cannot immediately migrate to full HTTPS. Due to both the slow adoption rate of HTTPS as well as an increasingly tangled web, many resources and services currently block the widespread deployment of HTTPS.

To understand what organizations are blocking HTTPS adoption, we looked at the external resources that are most commonly included over HTTP and are not yet available over HTTPS. As part of this analysis, we considered two types of blockers: direct blockers (specific HTTP-only resources that directly inhibit HTTPS adoption) and indirect blockers (services that include their own content over HTTPS, but load third-party content over HTTP). An example of a direct blocker is an analytics script that is only accessible over HTTP. An example of an indirect blocker is an ad provider whose JavaScript is served over HTTPS, but whose third-party ad content is served over HTTP.

The direct blocker that prevents the most sites from upgrading is a piece of JavaScript loaded from b.scorecardresearch.com, which belongs to comScore, a popular tracking and analytics platform. The script is included on 4.5% of the top 1M, 12% of the top 100K, and 27% of the top 10K sites. There are two other services that additionally block more than 10% of the Top 10K: Casale Media (22% of top 10K) and an Alexa Tracker (11% of top 10K). Beyond these, there is a long tail of additional services that remain inaccessible over HTTPS. We categorized the top 25 blockers, finding that 40% are ad providers, 32% are analytics services, and 8% are social media plugins. We list the services that are blocking the most sites from migrating to HTTPS in Figure 10.

In analyzing direct blockers, we also find evidence of blocker loops—these are sites that are currently all HTTP-only, but are interdependent on one another. In other words, no site in a loop can upgrade to HTTPS unless every other site in a loop upgrades as well. We find 5,872 blocker loops in the HTTP sites on the top million, but on closer inspection, we find many sites that are distinct by public-suffix are owned and operated by the same entity (for example, oneindia.in vs. oneindia.com). We remove sites that follow this pattern and find a total of 5,112 direct blocker loops in the top million sites.

We calculated the largest indirect blockers as the services that load HTTP-only content onto the most websites. As can be seen

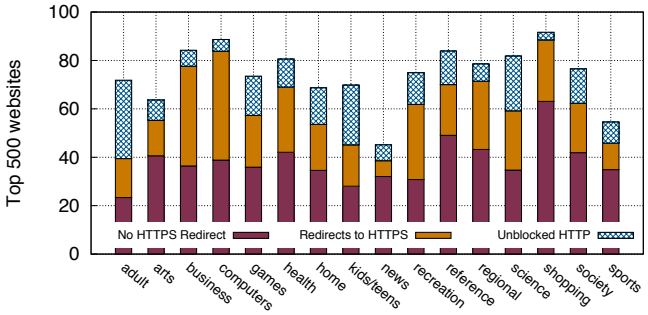


Figure 11: **HTTPS Adoption by Alexa Category** We show the HTTPS adoption sites that can immediately upgrade to HTTPS by Alexa Category. If a site supports HTTPS, we show whether the root page redirects to HTTPS by default or not. If a site is HTTP-only, we show the fraction of sites that can immediately upgrade to HTTPS. The remainder of sites are blocked by some resource not yet available over HTTPS.

Domain	T 1M	HTTPS	Domain	T 1M	HTTPS
googlesyndication.com	1.8%	71%	exoclick.com	0.8%	24%
pubmatic.com	1.3%	8%	openx.net	0.7%	9%
casalemedia.com	1.2%	14%	crwdcntrl.net	0.7%	5%
sharethis.com	1.2%	13%	exelator.com	0.7%	34%
turn.com	1.1%	4%	doubleclick.net	0.7%	99%

Figure 12: **Services that load third-party HTTP content** — We show the services that include the most third-party content over HTTP and how much content that service serves over HTTPS.

in Figure 12, the majority are ad providers. It is important to note that indirect blockers do not exclusively load content over HTTP. They sometimes load third-party content over HTTPS, but to drastically varying degrees. For example, the indirect blocker service that loads the most third-party HTTP content is Google (via googlesyndication.com). However, more than 70% of content loaded by googlesyndication.com is loaded over HTTPS and the service likely is not blocking sites from migrating to HTTPS. Unfortunately, all of the non-Google services in the top ten serve far less total content over HTTPS. For example, the second most popular indirect blocker, PubMatic, serves only 8% of third-party content over HTTPS. We note that none of these non-Google providers appear to match the protocol of the root website (i.e., always serve HTTP-based ads to HTTP-based sites), and we suspect that these ad providers are preventing sites from migrating to HTTPS.

Blocked Websites by Category There is significant variance in HTTPS deployment between different types of sites. For example, while 89% of shopping sites have deployed HTTPS, 61% of news sites remain HTTP-only (Figure 11). We also find that many blockers are category specific, for example, 72% of news sites block on comScore, 28% block on Turn, 26% on Casale Media, and 21% on Moat Ads.

Part of this difference is due to the fact that each category only has the top 500 sites, skewing blockers towards the services deployed on the most popular websites. This is similar to how we observe these popular services blocking a larger percent of the top 10K than the top 1M. comScore remains the single largest blocker in all but one category (shopping), but nearly 72% of news and 65% of art sites block on it, whereas it only blocks 4.5% of the full top million. In other cases, we find domain-specific blockers. For example, adult sites block on Juicy Ads, LongTail Ad Solutions, and Friend Finder Networks, but these do not appear on the global list. We list the five largest blockers for each type of site in Figure 13.

While the individual blockers we identify only affect a fraction of the top million, we find evidence that these third parties block the most popular sites from migrating. Different perspectives identify slightly differing sets of blockers, but a set of direct blockers appears consistently: comScore, Casale Media, Baidu, ShareThis, Turn, Moat Ads, Network Solutions, Stat Counter, Alexa, and AdRoll. We further note that a second class of *indirect* blockers exist, which may block websites from migrating by loading third-party content over HTTP, including PubMatic, Casale Media, Turn, OpenX, and Crowd Control. We strongly encourage these sites to prioritize upgrading to HTTPS and/or requiring their affiliates to use HTTPS, as they are currently preventing a large number of the most popular sites from upgrading.

6. RELATED WORK

There have been a large number of studies on the increasing complexity of the web [5, 10, 13, 16, 17, 25]. Most similar to our study, Butkiewicz et al. studied web complexity in 2011 [5]. We find that the web has changed dramatically over the past five years, and we describe the differences in Section 3.

Past work has studied the increased risk associated with including third-party content. Initial work in the drive-by-download area noted that in addition to webpage compromise, the inclusion of external objects inherently elevates the risk profile of a website [6, 27, 28]. More recently, Nikiforakis et al. investigated the impact of external JavaScript providers and the vulnerabilities that may arise from included third-party content [25].

A number of recent papers have focused on the privacy implications of online tracking [17, 22, 31]. Hand in hand with such studies are proposed defenses against malicious content: Levy et al. and Arshad et al. present methods that detect and block malicious external content [2, 18].

Performance is an additional consequence of increased web complexity. A number of previous studies use this landscape to study the effectiveness and performance of web caching [4, 29]. While web caching is beyond the scope of this paper, certain caching techniques certainly influence decisions of where and how to properly load resources on the web [16, 35], and the effects of such decisions may introduce new security and privacy challenges. A particularly relevant piece of work in this domain is Polaris [24], which creates fine-grained dependency graphs in order to help prioritize object loads.

Most similar to our measurement platform is OpenWPM, a tool built by researchers at Princeton University [9]. Our measurements are similar in scope and scale to those reported in the Internet wide study done this year with OpenWPM [10], however, we implement our measurement from a different browser vantage point (Chrome vs Firefox), and analyze a different set of attributes.

7. CONCLUSION

In this work, we investigated the current state of website dependencies and the security challenges introduced by the increasing reliance on external services. We found that websites load nearly double the number of dependencies than five years ago and that only a negligible number of sites remain altogether dependency-free. Further, sites are increasingly interconnected: 64% of resources are now externally loaded and more than 90% of sites rely on content from other sites and networks. Worryingly, one third of the top million sites serve implicitly-trusted, unknown content from ad networks and other third parties — drastically increasing these sites' attack

Top Blockers	Sites	Type
News	439	
b.scorecardresearch.com	72.4%	T
cdn.turn.com	27.6%	A
ip.casalemedia.com	26.0%	A
dsum.casalemedia.com	25.5%	A
js.moatads.com	20.1	A
Arts	359	
b.scorecardresearch.com	64.6%	T
js.moatads.com	24.5%	A
dsum.casalemedia.com	19.4%	A
cdn.turn.com	17.5%	A
ip.casalemedia.com	13.9%	A
Business	347	
b.scorecardresearch.com	26.5%	T
dsum.casalemedia.com	12.1%	A
cdn.clicktale.net	8.6%	A
global.ib-ibi.com	6.6%	U
ds.serving-sys.com	6.3%	A
Shopping	317	
dsum.casalemedia.com	18.9%	A
b.monetate.net	16.1%	U
global.ib-ibi.com	11.7%	U
cache.dtmpub.com	11.7%	A
a.adroll.com	10.7%	A
Sports	279	
b.scorecardresearch.com	55.6%	T
dsum.casalemedia.com	23.7%	A
cdn.turn.com	19.0%	A
global.ib-ibi.com	16.1%	U
js.moatads.com	15.4%	A

Top Blockers	Sites	Type
Home	277	
b.scorecardresearch.com	59.2%	T
as.casalemedia.com	19.9%	A
global.ib-ibi.com	19.9%	U
ssum.casalemedia.com	19.1%	A
dsum.casalemedia.com	17.7%	A
Computers	273	
b.scorecardresearch.com	34.4%	T
dsum.casalemedia.com	18.7%	A
global.ib-ibi.com	12.5%	U
cdn.clicktale.net	12.5%	A
a.adroll.com	11.0%	A
Recreation	265	
b.scorecardresearch.com	30.1%	T
dsum.casalemedia.com	12.0%	A
global.ib-ibi.com	11.7%	U
cdn.turn.com	8.3%	A
as.casalemedia.com	7.9%	A
Games	221	
b.scorecardresearch.com	39.8%	T
dsum.casalemedia.com	30.0%	A
global.ib-ibi.com	22.6%	U
cdn.turn.com	22.6%	A
ip.casalemedia.com	20.4%	A
Adult	212	
b.scorecardresearch	9.0%	T
ads-a.juicyads.com	4.2%	A
p.jwpcdn.com	3.8%	A
graphics.pop6.com	2.8%	A
cloudfront-labs.amazonaws.com	2.8%	T

Top Blockers	Sites	Type
Teens	194	
b.scorecardresearch.com	43.3%	T
dsum.casalemedia.com	22.2%	A
global.ib-ibi.com	14.4%	U
cdn.turn.com	13.9%	A
ip.casalemedia.com	12.9%	A
Science	150	
b.scorecardresearch.com	41	T
dsum.casalemedia.com	31.7%	A
a.adroll.com	29.3%	A
ssum.casalemedia.com	29.3%	A
admin.brightcove.com	26.8%	A
Health	130	
b.scorecardresearch.com	43.8%	T
a.adroll.com	13.1%	A
edge.sharethis.com	10.8%	S
w.sharethis.com	10.8%	S
global.ib-ibi.com	8.5%	U
Reference	130	
b.scorecardresearch.com	18.5%	T
dsum.casalemedia.com	13.1%	A
www.infobel.com	12.3%	A
a.adroll.com	8.5%	A
as.casalemedia.com	7.7%	A
Regional	444	
b.scorecardresearch.com	37.2%	T
dsum.casalemedia.com	18.7%	A
global.ib-ibi.com	14.0%	U
cdn.turn.com	11.7%	A
tap2-cdn.rubiconproject.com	11.7%	A

Figure 13: **Top HTTPS Blockers by Alexa Category** — We show the top HTTPS blockers for each Alexa category. The foremost blocker across almost all categories is b.scorecardresearch.com, which belongs to comScore, an Internet market research provider. T is tracking, A is advertising, U is unknown.

surface. Not only does this tangle of dependencies introduce a host of security and privacy concerns, but the increased complexity is blocking websites from migrating to HTTPS. Nearly 28% of HTTP sites are blocked from upgrading to HTTPS and 55% of HTTP sites will not be fully HTTPS compliant until their dependencies migrate. In the most egregious case, comScore is blocking 27% of the Top 10K sites from upgrading. We hope that our findings motivate future research on security and privacy solutions for the web, prompt service providers to migrate to HTTPS, and encourage developers to consider the implications of serving third-party content.

Acknowledgments

The authors thank David Adrian, Richard Barnes, and Vern Paxson, for their help and feedback. We thank the exceptional sysadmins at the University of Michigan for their help and support, including Chris Brenner, Kevin Cheek, Laura Fink, Dan Maletta, Jeff Richardson, Don Winsor, and others from ITS, CAEN, and DCO. This material is based upon work supported by the National Science Foundation under awards CNS-1345254, CNS-1409505, CNS-1518888, CNS-1505790, CNS-1530915, and CNS-1518741, by the Department of Energy under DE-OE0000780, by a Google Ph.D. Fellowship, and by an Alfred P. Sloan Foundation Research Fellowship.

8. REFERENCES

[1] Alexa Internet, Inc. Alexa Top 1,000,000 Sites. http://s3.amazonaws.com/alexa-static/top-1m.csv.zip.

[2] S. Arshad, A. Kharraz, and W. Robertson. Include me out: In-browser detection of malicious third-party content inclusions. In *20th Conference on Financial Cryptography and Data Security*, 2016.

[3] R. Barnes. Mozilla security blog: Deprecating non-secure HTTP. https://blog.mozilla.org/security/2015/04/30/deprecating-non-secure-http/.

[4] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker. Web caching and zipf-like distributions: Evidence and implications. In *18th IEEE International Conference on Computer Communications*, 1999.

[5] M. Butkiewicz, H. V. Madhyastha, and V. Sekar. Understanding website complexity: measurements, metrics, and implications. In *11th ACM Internet Measurement Conference*, 2011.

[6] M. Cova, C. Kruegel, and G. Vigna. Detection and analysis of drive-by-download attacks and malicious JavaScript code. In *19th World Wide Web Conference*, 2010.

[7] Z. Durumeric, D. Adrian, A. Mirian, M. Bailey, and J. A. Halderman. A search engine backed by Internet-wide scanning. In *22nd ACM Conference on Computer and Communications Security*, Oct. 2015.

[8] Z. Durumeric, E. Wustrow, and J. A. Halderman. ZMap: Fast Internet-wide scanning and its security applications. In *22nd USENIX Security Symposium*, 2013.

[9] S. Englehardt, C. Eubank, P. Zimmerman, D. Reisman, and A. Narayanan. Openwpm: An automated platform for web privacy measurement, 2015.

[10] S. Englehardt and A. Narayanan. Online tracking: A 1-million-site measurement and analysis. In *23rd ACM Conference on Computer and Communications Security*, 2016.

[11] S. Farrell and H. Tschofenig. Pervasive monitoring is an attack. 2014.

[12] Google. Headless chromium. https://chromium.googlesource.com/chromium/src/+/lkgr/headless/README.md.

[13] S. Ihm and V. S. Pai. Towards understanding modern web traffic. In *11th ACM Internet measurement conference*, 2011.

[14] Internet Architecture Board. IAB Statement on Internet Confidentiality. https://www.iab.org/2014/11/14/iab-statement-on-internet-confidentiality/.

[15] E. Konigsburg, R. Pant, and E. Kvochko. Embracing HTTPS. http://open.blogs.nytimes.com/2014/11/13/embracing-https.

[16] B. Krishnamurthy and C. E. Wills. Cat and mouse: Content delivery tradeoffs in web access. In *15th World Wide Web Conference*, 2006.

[17] A. Lerner, A. K. Simpson, T. Kohno, and F. Roesner. Internet jones and the raiders of the lost trackers: An archaeological study of web tracking from 1996 to 2016. In *25th USENIX Security Symposium*, 2016.

[18] A. Levy, H. Corrigan-Gibbs, and D. Boneh. Stickler: Defending against malicious content distribution networks in an unmodified browser. 2016.

[19] Malwarebytes Labs. Large angler malvertising campaign hits top publishers. https://blog.malwarebytes.com/threat-analysis/2016/03/large-angler-malvertising-campaign-hits-top-publishers/.

[20] MaxCDN. BootstrapCDN security post-mortem. https://www.maxcdn.com/blog/bootstrapcdn-security-post-mortem/.

[21] MaxMind, LLC. GeoIP2 database. http://www.maxmind.com/en/city.

[22] J. R. Mayer and J. C. Mitchell. Third-party web tracking: Policy and technology. In *IEEE Symposium on Security and Privacy*, 2012.

[23] Mozilla Foundation. Public suffix list. https://publicsuffix.org/.

[24] R. Netravali, J. Mickens, and H. Balakrishnan. Polaris: Faster page loads using fine-grained dependency tracking. In *13th USENIX Symposium on Networked Systems Design and Implementation*, Mar. 2016.

[25] N. Nikiforakis, L. Invernizzi, A. Kapravelos, S. Van Acker, W. Joosen, C. Kruegel, F. Piessens, and G. Vigna. You are what you include: large-scale evaluation of remote javascript inclusions. In *19th ACM Conference on Computer and Communications Security*, 2012.

[26] T. C. Projects. Marking HTTP As Non-Secure. https://www.chromium.org/Home/chromium-security/marking-http-as-non-secure.

[27] N. Provos, P. Mavrommatis, M. A. Rajab, and F. Monrose. All your iframes point to us. In *17th USENIX Security Symposium*, 2008.

[28] N. Provos, D. McNamee, P. Mavrommatis, K. Wang, and N. Modadugu. The ghost in the browser analysis of web-based malware. In *1st Workshop on Hot Topics in Understanding Botnets*, Berkeley, CA, USA, 2007.

[29] M. Rabinovich and O. Spatscheck. *Web caching and replication*. Addison-Wesley, 2002.

[30] B. Riordan-Butterworth. Adopting encryption: The need for https. http://www.iab.com/adopting-encryption-the-need-for-https/.

[31] F. Roesner, T. Kohno, and D. Wetherall. Detecting and defending against third-party tracking on the web. In *9th USENIX Networked Systems Design and Implementation*, 2012.

[32] Z. Tollman. We're going HTTPS: Here's how WIRED is tackling a huge security upgrade. https://www.wired.com/2016/04/wired-launching-https-security-upgrade.

[33] W3C. Securing the web. https://w3ctag.github.io/web-https/.

[34] W3C. Subresource integrity. https://www.w3.org/TR/SRI/.

[35] X. S. Wang, A. Krishnamurthy, and D. Wetherall. How much can we micro-cache web pages? In *14th ACM Internet Measurement Conference*, 2014.

[36] W. V. Wazer. Moving the Washington Post to HTTPS. https://developer.washingtonpost.com/pb/blog/post/2015/12/10/moving-the-washington-post-to-https/.

[37] S. Yuan, J. Wang, and X. Zhao. Real-time bidding for online advertising: measurement and analysis. In *International Workshop on Data Mining for Online Advertising*, 2013.