HTTPS Adoption in the Longtail

Ariana Mirian
University of California, San Diego
amirian@cs.ucsd.edu

Christopher Thompson
Google
cthomp@google.com

Stefan Savage
University of California, San Diego
savage@cs.ucsd.edu

Geoffrey M. Voelker
University of California, San Diego
voelker@cs.ucsd.edu

Adrienne Porter Felt
Google
felt@google.com

ABSTRACT

HTTPS is widely acknowledged as a pillar of modern web security. However, while much attention focuses on the value delivered by protocol improvements, the benefit of these advances is gated by the breadth of their adoption. Thus, while the majority of web pages visited benefit from the confidentiality and integrity guarantees of HTTPS, this is contradictorily due to a minority of popular sites currently supporting the protocol. In this paper written in April 2018, we explore factors of HTTPS adoption on web sites more broadly. We analyze attributes of the Alexa top one million sites in August 2017 and categorize them into popular and "longtail" sites, in an effort to identify points of leverage which offer promise for driving further adoption of HTTPS. We find that hosting provider use and cost are factors that correlate with HTTPS deployment, while other promising indicators—such as site age, site freshness, and server software choice—provide ambiguous signals and are unlikely to offer useful points of influence.

1 INTRODUCTION

Security technologies, such as Transport Layer Security (TLS), provide concrete capabilities that protect users against potential threats. In particular, web sites that protect access via TLS (aka "HTTPS") can offer meaningful guarantees of confidentiality and integrity for all communications between their site and their visitors. While there have been a range of attacks against prior iterations of these protocols and against their implementations, there is wide agreement that HTTPS is one of the success stories of modern cryptography and network security technology.

However, like most communications technologies, the value of a protocol is innately tied to the breadth of its deployment. On the client-side, this is effectively a non-issue, as all major web browsers have supported TLS for many years. Similarly, all major web server systems have supported TLS for as long. Given these facts, it is therefore surprising that only a minority of Internet web sites will accept and correctly respond to HTTPS requests.

The core of the lack of adoption arises from the fact that HTTPS is an option, not a default. Configuring a server to support HTTPS is an individual decision for each site operator. Thus, every operator must realize that HTTPS is an option, determine that it is a valuable capability for their visitors, decide that encryption overhead is supportable with their existing systems and software, contract with a certificate authority (CA) to acquire a certificate, and, finally, develop the skill to configure the server appropriately. Operators using third-party hosting must further contend with the technology and business choices made by their service providers, and interactions with third-party advertisers and web search engines similarly can create additional complications.

The most popular web sites have been the quickest to adopt HTTPS. When scanning the 10,000 most popular sites (as ranked by Alexa) in August of 2017, over 60% provided HTTPS connectivity. These popular sites have a number of natural advantages, such as dedicated IT staff naturally funded by the advertising or e-commerce revenue that accrues with online popularity.

The popularity of these sites also attracts pressure from security-minded customers and advocacy campaigns. For example, Google publishes a transparency report that lists the various HTTPS aspects of the 100 most popular web sites, such as whether the sites work on HTTPS, have HTTPS on by default, and have a modern TLS configuration [39]. Similarly, there have been concerted efforts to move specific Internet sectors over to HTTPS as well. For example, Secure The News and Pulse are initiatives focused on transitioning and monitoring news and United States government sites over to HTTPS [25, 40]. In an effort to help popular sites transition to HTTPS, the Google Chrome team has published a variety of resources covering transition approaches and describing the experiences of large web sites in managing their HTTPS deployment [12, 35, 37]. Encouragingly, this focus has offered significant payoff because these popular sites capture the lion's share of Internet traffic. As a result, today the majority of web page loads associated with web content are transported via HTTPS [38].

However, these advantages do not scale to the "longtail" that comprises the vast majority of web sites in operation today. Indeed, these smaller sites (personal blogs, small businesses, community forums, etc., which might not be easily grouped together into one sector) represent much of the web's diverse content, but are less likely to offer the protection of HTTPS. Concretely, after excluding the top 10,000 most popular sites, the remainder of the top one million sites only have a 45% HTTPS adoption rate. Thus, the amount of individualized work necessary to increase HTTPS adoption further is only going to increase.

Because of this issue, we want to understand at scale whether HTTPS adoption decisions are entirely specific to individual sites or if there are segments of the site population more amenable to upgrading their security posture. In this paper, we identify correlates of HTTPS deployment—separately for popular and "long tail" sites—that provide insight into ways to effect further changes (or

This paper was originally written in April 2018, based on data collected from the Alexa Top 1 Million sites in August 2017.

alternatively, to understand the most significant barriers to adoption that deserve further attention). We evaluate several factors including:

- **Site age:** Are more recent sites, in terms of age and freshness, more likely to use HTTPS?
- **Server Software:** Does the type of server software used correlate with HTTPS adoption?
- **Hosting status:** Are sites operated by hosting providers more likely to offer HTTPS compared to self-hosted sites? How might this vary by provider?
- **Role of certificate authorities:** How do certificate cost and business offerings from hosting providers affect HTTPS deployment?

We focus on individual variables in the HTTPS equation as a first step in understanding at scale the HTTPS adoption hurdles web sites face. Moreover, we focus on variables that reflect degrees of site administrator involvement, like site age and freshness, as well as variables that a site administrator directly controls, such as server software, hosting provider, and certificate authority. Notably, we identify that key correlates of HTTPS adoption, particularly for less popular sites, are cost and free, automatic hosting provider support for HTTPS. Thus, HTTPS evangelists should consider focusing on working with leading certificate authorities (CAs) and hosting providers whose service offerings do not yet encourage broad HTTPS use.

2 BACKGROUND

The Hyper-Text Transport Protocol (HTTP), as originally designed in 1991, did not address security in any way. However, even before HTTP 1.0 was officially standardized in 1996, browsers and servers had started supporting a secure version of HTTP, tunneled within an encrypted session protocol, and `https://` came to identify the secure variant of HTTP. The basic goal and approach of HTTPS have remained the same to this day, despite the underlying technology having changed several times (from the original Secure Sockets Layer protocol to the multiple iterations of Transport Layer Security).

HTTPS guarantees communications confidentiality and integrity against an adversary who might spoof, intercept, or modify messages sent between a browser and web server. It can also guarantee domain authenticity (i.e., that the web site you are communicating with at `foo.com` can provide compelling evidence that it has rights to this domain name). These guarantees depend on both sides of the connection correctly implementing the protocol *and* on a trusted certificate authority (CA) properly validating the site and provisioning a certificate.

A web site operator seeking to offer HTTPS has several obligations. First, they must obtain a certificate, from a trusted certificate authority, that attests to their ownership of the domain on which the site is hosted. The CA, in turn, must perform some amount of due diligence to validate this claim. This can range from "domain validation" (i.e., being able to serve a particular file of the CA's choosing on the Web site) to more traditional documentary evidence of business incorporation and domain registration. Second, the site owner must configure their server to enable HTTPS connections using the certificate they have acquired. Since certificates expire, the web site operator is also obliged to periodically renew their certificate to ensure that it remains valid. Finally, the site operator must avoid loading any sub-resources over HTTP ("mixed content"), as any such resource would lack the security guarantees of HTTPS. Browsers commonly downgrade the security indicator for sites with mixed content, or prevent loading the resources entirely, potentially breaking the functionality of the site.

The complexity and costs of each of these steps can vary significantly from one environment to the next. For example, most CAs fund their operations by charging annual fees for certificates (which can run up to hundreds of dollars per year) while not-for-profit Let's Encrypt offers certificates for free. Some CAs and server software options are implicitly designed for highly sophisticated customers who have interest in configuring the details of their certificate or the choice of cryptographic protocols supported on their servers. Other environments prioritize ease of use and can provision certificates in a single step and expect users to abide by reasonable defaults. These issues can be further complicated for more sophisticated sites which may include significant legacy content and multiple tiers of servers working in concert to provide a service.

Finally, to incentivize site operators to adopt HTTPS, several browser vendors (notably the developers of Firefox and Chrome), have chosen to visually mark sites using HTTP as "not secure" (which is highlighted if a user enters text into the page). Moreover, Google has also announced that sites using HTTPS will be given ranking advantage in Google's search engine.

3 RELATED WORK

HTTPS has a long and rich history of research, both due to its importance in the web security ecosystem and its mechanical complexity. Many early studies around HTTPS focused on characterizing the certificate ecosystem, in an effort to understand what the problems were. In 2011, Holz et al. analyzed multiple sources of certificates to characterize the state of the X.509 public key infrastructure (PKI), and found a plethora of errors around the deployment of certificates, which provide a trusted foundation of HTTPS authentication [42]. In a follow up study, Durumeric et al. took a wider view of the HTTPS ecosystem using Internet-wide scanning, which exposed even more technical problems in the web's PKI [19]. Particularly focusing on the use of HTTPS in Content Delivery Networks (CDNs), Liang et al. empirically established that CDN customers delegate their credentials—either by direct sharing of private keys, or using the Subject Alternative Name (SAN) extension [47]. Building on this observation, Cangialosi et al. perform a large-scale study of HTTPS certificate hosting, and demonstrate that popular sites routinely share their private keys with CDNs to benefit from their economies of scale [14]. Our work focuses on a tangential issue—exploring what factors influence the adoption of HTTPS among *less popular* sites—but also explores hosting choices as one of these factors.

Other studies have focused on vulnerabilities in HTTPS implementations and how the broader Internet community addresses these issues. For example, the Heartbleed vulnerability affected an estimated 24–55% of popular HTTPS sites [20]. A study by Durumeric et al. looked at the effectiveness of notifications in the aftermath of this massive vulnerability and found that notified sites patched their servers 50% more than non-notified sites. In spite of these relative effects, Zhang et al. examined overall certificate revocation and reissuance in the aftermath of Heartbleed and found that sites were slow to revoke and reissue, and when sites did reissue certificates, it was often with the same vulnerable keys [54]. Heartbleed is not unique among massive vulnerabilities and other research has identified a range of similarly widespread security vulnerabilities (e.g., LogJam, Drown) [1, 11]. Independent of such large-scale vulnerability disclosures, there have also been a variety of studies on reasons why certificates are invalid as well as certificate revocation in the wild [15, 49, 54].

In addition to the mechanical exploration of HTTPS and how vulnerabilities affect the ecosystem, a number of researchers have focused on end-user centered questions related to HTTPS signaling and efforts to improve user adherence to HTTPS warnings. Akhawe et al. looked at the difference in HTTPS warning click-through rates (the rates at which users ignore HTTPS warnings) from both the Firefox and Chrome perspective [3]. The vast difference in click-through rates indicated a significant difference in the error signaling of the two browsers, and that Firefox was more effective in protecting end users. Following this work, Felt et al. created a more effective warning for Chrome, preventing 30% more users from clicking through HTTPS warnings [22]. Akhawe et al. also built on the previous work by looking at HTTPS errors from a network perspective [2], and categorized the types of errors that users see. Finally, work has improved the HTTPS indicator shown to users, by adding the word "Secure" next to the green lock that major browsers display with a valid HTTPS site [24].

Most similar to our work is that of Felt et al. which focused on how to best measure HTTPS adoption via different methods and datasets [23]. In this study, the authors identified the discrepancy in adoption rates between popular sites and "longtail" sites, and provided the motivation for our work. Kumar et al. examined the relationship between third-party resources and the expanded attack surface that HTTP links on an HTTPS page introduce [46]. Also related, Krombholz et al. explored HTTPS deployment in a controlled setting to understand the kinds of challenges faced by those without dedicated technical staff [45]. By contrast, we explore similar questions but using large-scale measurement data to understand what factors correlate with HTTPS adoption across a million sites.

4 DATA

We evaluate HTTPS adoption among the sites on the Alexa Top 1 Million list as of August 31, 2017 [4]. Of these sites, we distinguish between the most popular 10,000 sites as "top sites" and the remaining 990,000 sites as the "longtail". We use the Alexa list since it is a public proxy for web site popularity ranking. Moreover, we consider the top sites as the most popular 10,000 sites based loosely on previous work by Felt et al. [23], where the most popular sites were defined as the top 100 of the Alexa Top 1 Million. We take a more liberal approach in our definition of top sites to better include a wider range of behaviors. We these categories in mind, we found that 60% of top sites support HTTPS whereas only 45% of longtail sites support it, a significant difference.

To determine the HTTPS status of these sites, we scan them using `pshtt`, an open source tool that specifically checks for HTTPS support [51]. Originally, `pshtt` was used to scan US government sites; we modified it to scan more broadly (e.g., we added a check for bad Subject Alternative Name (SAN) and changed the timeout mechanism to be more forgiving). `pshtt` scans a site on four endpoints (`http` or `https`, and with and without the `www` subdomain) and records the headers provided by each endpoint, as well as the HTTPS status of the site. `pshtt` also attempts to determine the "canonical URL" for a site, or the endpoint that a user is likely to be redirected to. For example, many sites will redirect their non-`www` endpoint to their `www` endpoint, or their `http` endpoint to `https`; `pshtt` makes a guess based on the data that it collects.

For a single-day snapshot, we ran `pshtt` over the Alexa Top 1 Million sites on multiple Google Cloud Compute Engine instances on August 31, 2017. Unless otherwise stated, we use the `pshtt` results on the canonical URL endpoint for each site domain. We also use the Alexa 1 Million Snapshots from Censys [18] to gather certificate information for the Alexa domains. Censys provides a variety of different scanning data sets, in particular parsed certificate data from web sites over time. We use the Alexa snapshots for every month from August 2017 backwards in time to August of 2015 (when these snapshots first became available on Censys). We also use a snapshot from December 17th, 2017 as a comparison to our `pshtt` scan in August to evaluate certificate persistence over time.

5 SITE AGE AND FRESHNESS

The age of a site—when a site first went live on the web—is a tantalizing feature for correlating with HTTPS adoption. Are older sites more or less likely to support HTTPS, particularly in the longtail? Being old, perhaps such sites have inertia against adopting new features. Alternatively, perhaps long-term familiarity with operating a site makes it more likely that an operator will adopt new features. At the same time, the web ecosystem has increasingly made it easier for new sites to use HTTPS, such as with streamlined tools from certificate authorities or even automated HTTPS support from popular hosting providers. Any correlation between site age and HTTPS adoption can help determine whether adoption efforts should be focused on newer or older sites.

Similarly, how fresh a site's content is—the last time a site updated the content it serves—could also correlate with HTTPS adoption. If a site is not relatively fresh, for instance, perhaps the site operator is less likely to spend the time and effort to adopt new features such as HTTPS since they are not updating its content.

In this section, we explore both of these features as they relate to HTTPS adoption, first using various methods for estimating the ages of sites and then using an automated method for estimating when a site was last updated.

5.1 Site Age

There is no straightforward methodology for determining when a site first went live, and hence how old it is. As a result, we explore three distinct methods for estimating site age: two manual techniques on a small set of sites and one automated technique on a much larger set.

Domain WHOIS. As a first approach for estimating site age, we use two fields from WHOIS records for site domains: the "Creation Date" field reflects when the domain was created, while "Updated Date" reflects when certain details of the WHOIS record (such as contact information) are updated.

We download current WHOIS records from RiskIQ and bin their creation and update dates by year. RiskIQ provides WHOIS records

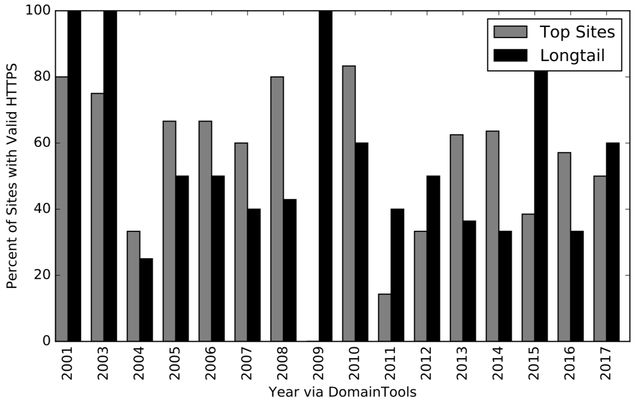

Figure 1: Domaintools sites bucketed by year and compared between top sites longtail sites. We examine sites from our 2017 dataset as they evolve over time.

at a higher rate limit than scraping ourselves, and we use it to obtain WHOIS records for a random subset of our domains: we download nearly 200,000 WHOIS records, which cover 67.3% of the top sites and 17.5% of the longtail. We look at site age from two perspectives:

- Using the creation date as the initial year for the domain, ignoring any updates to the record. However, domains can transfer ownership over time, much later than their original creation, so the creation date can be inaccurate.
- Using the update date as the initial year instead, under the assumption that the update date corresponds to an ownership change (e.g., a domain could be bought at auction, changing ownership). Of course, for some registrars, minor changes to WHOIS fields will also cause the update date to change (e.g., changing the technical contact), so the update date has inaccuracies as well.

Wayback Machine. We use the Wayback Machine (`archive.org`) as another method for estimating site age. The Wayback Machine preserves site content over time by crawling and storing periodic snapshots of sites across the web [43]. We select at random 100 top and 100 longtail sites and manually examine the landing page of the site for every year that the WayBack Machine has a snapshot, starting in August of 2017 and going backwards in time.

We define the initial year of a site as the year where the landing page has a significant change in purpose, a change that indicates that the site has very likely changed ownership. For example, `kayak.com`, a site in our sample of top sites, changed from a kayaking blog to a travel site, strongly suggesting a change in ownership. We also count when a domain switches from a parking page to actual content as a change in ownership. Sites that change cosmetically, though, do not indicate a change in site ownership. If the site does not have a significant content change, then we use the earliest year of the Wayback Machine records as the initial year for that site.

DomainTools. As a third method we use historical WHOIS records from DomainTools. Using the same random sample of 100 top and longtail sites (a total of 200 records), we manually analyze changes in the WHOIS records for the sites. If a record indicates a change in ownership, we consider the timestamp of that record as the initial year for that site. We do not consider minor changes, such as contact email or registrar changes, as a change in ownership. If the domain uses WHOIS privacy, but the "Update Date" in the record does not change, then we also do not consider that a change in ownership.

Results. Across the methods for estimating site age, we consistently find little dependence between the age of a site and whether it supports HTTPS. Figure 1 shows the results from DomainTools, where we manually examined historical WHOIS records for ownership changes. For each year, the graph shows the percent of top and longtail sites that appeared that year and supported HTTPS in August 2017. Visually there is no clear pattern, and graphs for the other two methods are similar. (Note that due to our sampling, no top sites fell into the 2009 bucket.)

More formally, we compute the *mutual information* of these two variables. Mutual information is useful because it means knowing information about one variable provides information about the other. It also conveniently has a conditional form incorporating a third variable, which we use below. In our case, mutual information (MI) measures the mutual dependence between the initial year a site appears on the web (site age) and whether it currently supports HTTPS. MI is 0 if the variables are completely independent: knowing the age of a site does not provide any information about whether the site supports HTTPS (and vice versa). MI is 1 if they are completely dependent: knowing site age deterministically predicts HTTPS support. For each of the site age estimation methods, we compute MI separately for the top sites and the longtail sites. These results are shown in the middle two columns of Table 1.

Across all methods, the MI for top sites is consistently higher than longtail sites; there is slightly more dependence among top sites and site age than with longtail sites. But the MI values for both kinds of sites are still relatively low—in both cases knowing the age of a site provides marginal information about whether a site supports HTTPS. In contrast to the other methods, MI is essentially zero when using the Update Date field from WHOIS records as site age. We therefore conclude that Update Date is not useful for estimating site age.

Combining all three variables, we also compute conditional mutual information where we condition off of whether a site is top or longtail. The conditional MI essentially answers the following question: if we know whether a site is a top site or a longtail site as a precondition, does knowing the age of a site give us any more information about whether the site supports HTTPS? The conditional MI is near 0 if the answer is negative: knowing site age does not give us any more information about whether the site supports HTTPS. Table 1 shows the conditional MI results for the site age estimation methods in the last column. The very low values indicate a negative result: if we already know that a site is top or longtail, then also knowing its age does not give any additional information about whether it supports HTTPS.

5.2 Site Freshness

To estimate how fresh a site's content is (when a site's content was most recently updated) we use the "Last-Modified" header provided by the server when scanning the site using `pshtt`. Originally created to make caching more efficient, this header is an optional self-reported field. In our data set, many sites do return this header: 19% of top sites and 21.5% of sites in the longtail. We remove clearly invalid last-modified times (e.g., 0.95% of top sites with this header

Age Estimation Method	Top Sites MI(site age, HTTPS)	Longtail Sites MI(site age, HTTPS)	MI (site age, HTTPS | top or longtail)
RiskIQ WHOIS Creation Date	0.20	0.003	0.01
RiskIQ WHOIS Update Date	0.003	0.001	0.008
Wayback Machine	0.24	0.18	0.07
Domaintools Historical WHOIS	0.16	0.13	0.07

Table 1: For the various methods for estimating site age, the middle columns show the mutual information (MI) values between site age and whether a site supports HTTPS. The last column shows conditional MI, where the MI is conditioned off of knowing whether a site is top or longtail.

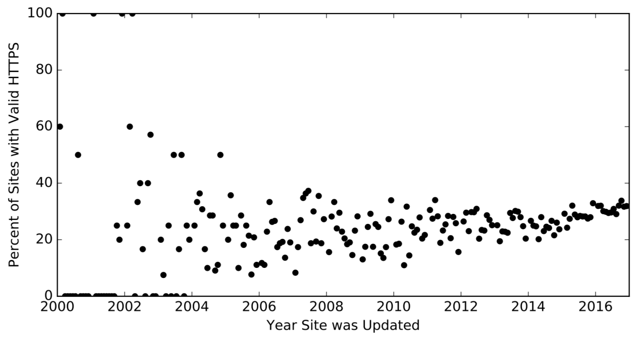

Figure 2: Scatter plot of the time that the content of a site was last updated (site freshness). Sites are binned into monthly buckets dating back to the year 2000, and the y-axis shows the percentage of sites in each bucket that support HTTPS.

and 1.6% of the longtail sites with this header report the Unix Epoch time from 1970). Of the sites that have this header, we exclude 1.2% of the top sites (0.002% of all top sites), and 0.12% of longtail sites with the header (0.0003% of all longtail sites).

For the sites providing this header, we group them by month and year of last modification. Figure 2 shows a scatter plot for the longtail sites grouped by month. For the sites that were last updated in a particular month and year, the graph shows the percentage of those sites that support HTTPS. Note that the vast majority of sites are relatively active and have been updated in the eight months preceding our crawl: 83.3% of longtail sites were last modified in 2017 (and 90.9% of top sites). So the high variance among points in the early years is due to having very few sites in each bucket. In recent years, the points cluster closely and suggest an upward trend.

However, as with site age, we also compute mutual information between when a site was last updated (how fresh it is) and whether it supports HTTPS. The conditional MI is just 0.02. Across all of the MI scores, the freshness of a site and whether it supports HTTPS are essentially independent (knowing the freshness of a site gives no information about whether it supports HTTPS).

5.3 Discussion

For all methods estimating site age, the mutual information between site age and HTTPS adoption is low, showing that site age is not a good indicator for HTTPS adoption. While we might have hoped that newer sites would be much more likely to be adopting HTTPS given increased awareness and support for security features, we found no strong dependence between the two variables. Large fractions of new sites that have come online even recently have not adopted HTTPS.

6 SERVER SOFTWARE

An important aspect of operating a site is the choice of server software, as an administrator individually managing a server must obtain and install a certificate, adding to the burden of adopting HTTPS. To add to this process, configuring HTTPS for a site depends upon which server package is used, and so the ease with which servers support HTTPS configuration could impact HTTPS adoption. Fortunately, some servers make this step easier. For example, Let's Encrypt provides software called certbot to make the process of obtaining and installing a certificate turnkey (i.e., a straightforward command line invocation). Currently, certbot only works with four server platforms: Apache, NGINX, HAProxy, and Plesk [21]. Without certbot, using Let's Encrypt takes more steps.

Since a site administrator's choice of server software can make HTTPS adoption easier or harder, we categorize top and longtail sites by the web server software platforms they use, and examine to what extent there is a dependence with HTTPS adoption. If a web server platform has certbot compatibility and better HTTPS adoption, then mechanisms like certbot can make HTTPS deployment easier.

We use the `pshtt` data to analyze the web server software that sites use. The `pshtt` scanner collects headers from the sites it scans. One of these headers is the "Server" header, which indicates the software package that the responding server uses. We fuzzy-match using the values of the Server header to categorize sites into the server software used. For example, `nginx/1.10.2` and `nginx/1.10.3` both group into the same bucket "nginx".

Most of the sites in our data set have a Server header: 85% of top sites, and 91% in the longtail. Of these, Figure 3 shows the most popular web server platforms they use and, for each server platform, the percentage of sites in the top and longtail that support HTTPS. Across all popular server platforms, servers in the longtail consistently have lower HTTPS adoption than top sites. Indeed, the two server platforms that Let's Encrypt's certbot supports, NGINX and Apache, together comprise over 50% of the servers in our data set. Yet, even with turnkey certificate management, servers in the longtail noticeably trail top sites in adoption. As another example, Caddy also supports turnkey HTTPS [13], yet fewer than 100 sites in our data set use it.

As further confirmation, we also compute mutual information between whether Let's Encrypt has turnkey support for the server

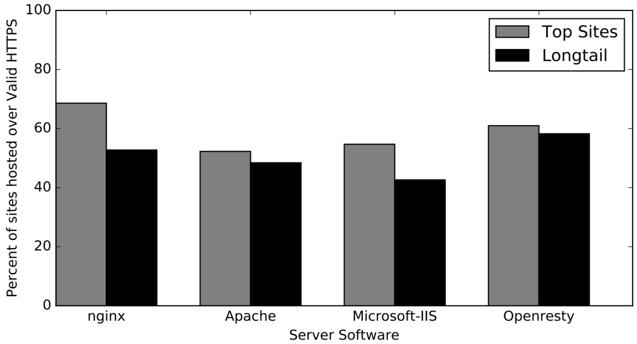

Figure 3: Popular server software platforms among top and longtail sites. For each platform, the bars show the percentage of top and longtail sites that support HTTPS.

software on a site, and whether the site supports HTTPS. The MI values are nearly 0 (0.002 among top sites and 0.003 in the longtail), as well as the conditional MI predicated on knowing whether a site is top or longtail. The MI results indicate no dependence between server software and a site using HTTPS.

Discussion. Server software is not an indicative factor for HTTPS adoption. The mutual information between server software and HTTPS is negligible, and all major server platforms consistently have lower HTTPS support in the longtail than in the top. A server's presence in the longtail is much more predictive of supporting HTTPS (or not) than the server platform used.

7 HOSTING PROVIDER

Hosting providers and content delivery networks (CDNs) serve an important role in the web ecosystem, particularly for owners of longtail sites. Longtail sites are inherently smaller operations which, unlike top sites, may even be operated and maintained just by individuals. Such individuals need technical expertise to maintain their own server, whereas provider services make maintaining a site accessible for anyone. As a result, hosting providers have a significant role in the deployment of HTTPS. If a hosting provider makes HTTPS less accessible to their customers, such as with higher costs or shifting the burden of configuration to users, their customers may be discouraged from using it. While many details of site deployment are abstracted from the administrator in this case, they must still choose a hosting provider, and this choice can have significant repercussions on HTTPS adoption.

We start by mapping top and longtail sites to the hosting providers they use, identifying the most popular providers and the degree of HTTPS adoption among them. We then register accounts and create web sites at the most popular providers for longtail sites, highlighting how well providers support even unsophisticated users in adopting HTTPS for their sites.

Top Sites			Longtail		
Provider	% Sites	% Valid	Provider	% Sites	% Valid
Cloudflare	17.4%	77.3%	Cloudflare	9.4%	80.6%
Amazon	9.1%	70.5%	Amazon	5.8%	60.0%
Google	2.8%	70.7%	OVH	3.4%	49.2%
Akamai	2.6%	66.9%	Google	2.8%	59.2%
OVH	1.8%	50.6%	GoDaddy	2.4%	23.8%
Chinanet	1.7%	38.6%	Hetzner	1.3%	42.8%
Fastly	0.9%	78.0%	Digital Ocean	1.3%	50.7%
Alisoft	0.9%	46.1%	Alisoft	1.2%	16.0%
Incapsula	0.8%	81.3%	Unified Layer	1.1%	32.0%
Microsoft	0.7%	48.6%	Linode	1.1%	48.5%
Self Hosted	4.7%	44.1%	Self Hosted	2.4%	25.4%

Table 2: Top ten providers based on IPWhois registrants.

7.1 Popular Providers

We use IP WHOIS records to identify providers for the domains in our site data set.[1] We crawled the DNS A record of each site's domain using ZDNS [53] to obtain its IP address; if the lookup returned a list of IP addresses, we use the first IP address in that list. We aggregated the set of IP addresses into /24 subnet granularities, and used the python IPWhois library on each /24 [44]. Since Regional Internet Registrys (RIRs) use multiple formats, we use the following fields, in order, as the name of the owner of the IP: registrant contact name, network name, and NIT nets name. Not all sites mapped to an IPWhois result with a name associated with it: 18% of top sites did not have a name mapping, and neither did 13% of the longtail. We exclude these from the provider analysis.

Overall, this method identifies many providers, but not surprisingly a small number account for much of the distribution with the remaining forming a long tail. The top sites map to 2,140 providers and the longtail sites map to 53,064. But the five most popular providers account for 34% of top sites, and the 20 most popular providers account for 38% of longtail sites.

Table 2 shows the most popular providers for the top and longtail sites according to the number of sites mapped to that provider, and the percentage of those sites that support HTTPS. Recall that HTTPS adoption among top sites is 60%, and the table shows that sites on major hosting providers such as Cloudflare, Amazon, and Google account for much of it. Yet, top sites hosted on other providers, such as Chinanet, Alisoft and notably Microsoft, are well below the average levels of HTTPS. Encouragingly, while HTTPS adoption among longtail sites is just 45%, adoption among longtail sites on the same major hosting providers as top sites is quite high (notably Cloudflare at 80.6%).

Figure 4 more explicitly compares the use of HTTPS between top and longtail sites for each hosting provider. It includes just those providers that appear in both top and longtail lists in Table 2. Although they vary considerably in terms of HTTPS adoption, some providers have consistent HTTPS adoption for both top and longtail sites (Cloudflare, OVH), while the others have noticeably higher HTTPS adoption for top sites. In the next section we explore whether features of the service can explain these differences.

[1] We also experimented with using the Autonomous System of a domain's IP address for additional provider information, but it did not contribute much beyond just WHOIS.

Hosting Provider	Type of Service	Percent Valid HTTPS Top Sites	Percent Valid HTTPS Longtail	Free HTTPS	Automatic HTTPS	Free Custom Certificate	Custom Domain	Can Upload Certificate
Cloudflare	CDN	77.3%	80.6%	★	★		★	
Amazon EC2*	VPS	70.5%	60.0%	★		★	★	★
Amazon Elastic Beanstalk	Hosting			★		★	★	★
Amazon Wordpress Hosting	Hosting			★		★	★	★
Amazon S3	Hosting					★	★	★
Amazon LightSail	Hosting			★		★	★	★
OVH	Hosting	50.6%	49.2%	★	★	★	★	
Google Sites	Drag/Drop	70.7%	59.2%	★	★			
Google Blogspot	Template			★	★			
Google App Engine	Hosting			★		★	★	★
GoDaddy Website Builder	Drag/Drop	12.5%	23.8%	★	★	★	★	
GoDaddy Website Hosting	CMS						★	★
GoDaddy Wordpress hosting	CMS						★	
Hetzner	Hosting	51.4%	42.8%	★		★	★	★
Digital Ocean	VPS	51.4%	50.7%	★		★	★	★
Linode	VPS	31.1%	48.5%	★		★	★	★
Self Hosted		44.1%	25.4%					

Table 3: Certificate features of the popular providers hosting longtail sites. "Free HTTPS": included in the package or does not require an additional fee. "Automatic HTTPS": HTTPS is enabled for a domain once the site starts running. "Free Custom Certificate": the service can provide a free custom certificate, again for no fee. "Custom Domain": the service allows custom domains to be used for the site. "Can Upload Certificate": customer can upload a custom certificate. *EC2 has a small surcharge per HTTPS connection.

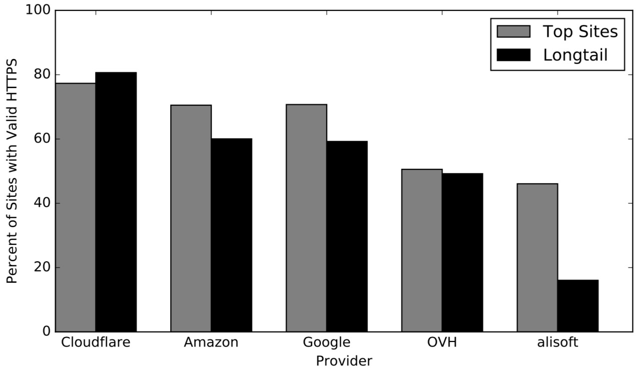

Figure 4: Top hosting providers for both top and longtail sites. Cloudflare and OVH sites have similar HTTPS adoption regardless of whether they are top or longtail sites.

Finally, we note that using IPWhois to map sites to hosting providers does have limitations. For example, Unified Layer does not provide customer hosting itself, but it does provide infrastructure for services that do. After experimenting with various methods for hosting provider identification, our conclusion is that improving the accuracy of large-scale hosting provider mappings for sites could be a separate project unto itself.

7.2 Customer Experience

From the provider breakdown, we see that a large portion of the sites are concentrated on a handful of major hosting and content providers, and that HTTPS adoption varies considerably among them. One factor for differences in HTTPS adoption could be the ease with which providers support customers in using HTTPS and configuring it properly. If a provider enables a customer to create a site and automatically and freely enable HTTPS, the provider substantially reduces the barrier for using HTTPS, particularly for unsophisticated site owners. In this section, we evaluate the different features these hosting providers support for HTTPS configuration, and determine if some providers make it easier than others to use HTTPS and configure it properly.

To perform this analysis, we engaged with eight longtail providers as a "longtail customer."[2] We registered an account and performed the steps necessary to create a web site using HTTPS at each of the providers. When providers had multiple services (e.g., Amazon EC2, Wordpress, Beanstalk, etc.), we created a separate site with each service. When they had multiple pricing tiers, we signed up for the least-expensive tier that provided HTTPS. For configuring the site domain, we tried both the service default (e.g., a subdomain of `wordpress.com`) as well as using a custom domain, if possible. For using HTTPS on our site, most providers encouragingly provide it for free; if they charged, then we paid the fee. Moreover, some providers enable HTTPS automatically; if they did not, then we performed the steps necessary to enable HTTPS. For configuring HTTPS, we used the default certificate option (e.g., a shared SSL certificate), but also tried acquiring a custom certificate for our site domain through the provider, as well as uploading our own certificate we created externally from the provider. If custom certificate configurations required a fee, we paid it to evaluate the difficulty of using this option. We signed up for these services between December 2017 and January 2018, with the exception of GoDaddy Website

[2] We did not sign up for Alisoft; as a China-based provider, they require an Internet Content Provider license, a "legal and mandatory requirement for all websites hosted on a server within the People's Republic of China". We also did not sign up for Unified Layer since, as discussed in Section 7.1, it does not sell hosting directly.

Hosting and GoDaddy Wordpress hosting, which we engaged with in April of 2018. We note the time frames as some of these services have changed since our testing; for example, Google Blogger now provides custom domain support for HTTPS.

As a last category, we also identify sites that are self-hosting. We use a manual approach based upon organizational information in WHOIS records. We take the top 100 organizations from WHOIS, by weight, for the top sites and manually classify them as whether they appear to be self-hosted sites (e.g., not using a hosting provider or CDN service to serve content).

Table 3 summarizes our results for creating and configuring sites using HTTPS on the various providers and their services. From these results we make a number of overall observations.

- Encouragingly, nearly all providers support using HTTPS for free. The understandable exceptions are the virtual private server providers (Digital Ocean and Linode), which provide virtual machines to customers. Amazon EC2, the other VPS service, nominally supports HTTPS for free, but in practice charges a higher per-connection fee for VMs that use it. The free tier of GoDaddy Website Builder does not support HTTPS, and GoDaddy's other service offerings charge a fee, which together could explain the very low HTTPS adoption of sites on GoDaddy.
- Three of the five providers with the highest HTTPS adoption rates (Cloudflare, Google and OVH) automatically enable HTTPS without needing customer action, either for all new sites (Cloudfare and OVH) or for new sites on a subset of their services (Google Sites and Blogspot; App Engine requires customer configuration).
- Recall from Figure 4 that Cloudflare and OVH had similar HTTPS adoption rates for both top and longtail sites. They are also the only providers that automatically enable HTTPS without exception (e.g., Google App Engine does not).
- Self-hosted sites have substantially lower HTTPS adoption rates than sites hosted on providers, particularly in the longtail.

We use the remainder of the section to detail our experiences with each provider. The details provide additional context for understanding the various services that providers offer, and their various fees, certificate options, etc.

Amazon. Amazon provides many different options for hosting a web site: EC2, Elastic Beanstalk, Wordpress Hosting, S3, and Lightsail [6–10]. For all use cases but EC2, certificates are free, but the cost of each connection over HTTPS is slightly higher than HTTP ($0.0025 per connection) [5]. With EC2, Amazon provides a virtual machine. Users can get a certificate through Let's Encrypt for free, but have to configure it themselves.

Cloudflare. Cloudflare is a global hosting provider and content delivery network [16]. Emulating a longtail customer, we use their free tier, which provides a free shared SSL certificate that is automatically enabled. However, if a customer wants to use their own certificate, there are two paid options: uploading a custom certificate after upgrading to their business tier, or purchasing a dedicated SSL certificate directly from Cloudfare ($5/month).

Digital Ocean and Linode. DigitalOcean and Linode are purely VPS providers [17, 48]. As with Amazon EC2, they offer virtual servers with various resource configurations. As a result, enabling HTTPS requires more advanced users who can figure out how to get a certificate via other means on their own, such as via Let's Encrypt.

GoDaddy. Godaddy offers three hosting options, and we try all three of them [26]. First, Website Builder has multiple tiers of increasing cost [30]. The bottom most tier, unfortunately, does not provide an SSL option whatsoever. The second tier, business, provides a certificate for sites under `*.godaddy`. If using a custom domain, then you can also get a custom certificate for free, but doing so requires the domain to be under GoDaddy's control. Users cannot upload their own certificates.

Website Hosting provides a certificate installer that automatically performs certificate setup and management [31]. It also has a higher cost—either purchase a certain number of months, or a certificate for $75—but it allows users to upload their own certificates.

Wordpress Hosting provide a wordpress management system on GoDaddy infrastructure [32]. Similar to Website Hosting, users can get a certificate for a fee, but does not support custom certificates.

Google. Google provides a variety of hosting services, and we used the three that longtail customers are most likely to use. Google Sites offers "classic" or "new" modes [36]. They differ in the domain used, but both use HTTPS by default and lead to `googlesites.com`.

Blogspot is a drag-and-drop blog and site creator [34]. It has two domain options: either create a subdomain of `blogspot.com`, or use a custom domain linked via Google Domains. When creating a `blogspot.com` subdomain, it automatically uses HTTPS. However, when using a custom domain, Blogspot warns that HTTPS is not currently available with that option.

Google App Engine allows users to upload and deploy apps on Google infrastructure [33]. App Engine has a certificate manager that works with Let's Encrypt and removes the hassle of certificate management. Users need to prove they own the domain and App Engine will take care of the rest of the configuration, but users do need to initiate this process by navigating to a panel.

Hetzner. Hetzner is a provider based in Germany [41]. We registered with their web hosting service using our own custom domain. With a custom domain, users can request a certificate for free from Hetzner, as long as the domain's A record points to Hetzner.

OVH. OVH is a global hosting provider based in France [50]. We registered as if we were a French customer; OVH services differ depending on which country you are located in, and offers the most services to French customers. OVH offers four hosting tiers; we use the lowest tier, Kimsufi (1.49 euro/month), which provides a domain name, storage, and email. When signing up with their hosting plan, users can choose among Wordpress, Joomla, Drupal, or Prestashop (we chose Wordpress). Once the domain was initialized, we were given access to the Wordpress managing system and a Let's Encrypt certificate was already installed for our site.

7.3 Discussion

Hosting providers do play a significant role in the HTTPS adoption of customers sites. Leading providers in the space, such as Cloudflare, offer both free certificates as well as automatically enabling HTTPS without the need for any customer action, features that provide a clearly compelling combination for impacting HTTPS adoption. To the extent that other hosting providers can be motivated to offer the combination of these features, in particular automatically enabling HTTPS beyond just offering free certificates, such efforts could have a significant impact on HTTPS adoption.

To further the point, if the top five providers among top sites streamlined moving their customer sites fully over to HTTPS, then HTTPS adoption among top sites would increase from 60% valid HTTPS to 75%. The longtail would require more outreach, but if the top 20 providers among the longtail did the same, then longtail HTTPS adoption would increase from 40% to 69%.

8 CERTIFICATE COST

At a high level, the larger, popular sites at the top will have more resources to devote to site maintenance and security than longtail sites. Top sites may have a specific team and budget devoted to their web site, while longtail sites might not even have a dedicated person, let alone a team, to manage their site. As a result, the cost of certificates may be dwarfed by other costs for top sites and become an insignificant factor, yet may still be an impediment for longtail sites. Indeed, certificate authorities themselves were concerned that certificate cost would be a factor to their customers. When Let's Encrypt opened to the general public, there were multiple CAs (some of which also provide hosting services) that marketed their more costly services as more secure and safer, in an effort to avoid losing customers to Let's Encrypt; Comodo even tried to trademark the name "Let's Encrypt" [27, 52].

In this section, we examine whether certificate cost is an impediment for HTTPS adoption in the longtail. We examine the use of certificate authorities and Let's Encrypt from three perspectives: popularity and prevalence, certificate authority migration, and certificate validation. We first look at popularity and prevalence to see if Let's Encrypt is a more popular certificate provider in the longtail, which is one indicator that cost is a factor for longtail site owners. We then look at CA migration to see if longtail sites are more likely to migrate to Let's Encrypt than top sites; if a site moves CAs, then there is some factor, such as cost, that provides an incentive to move instead of remaining with the original CA, where a site owner already has familiarity with setup and cost. Finally, we look at the type of certificate validation for the certificates in the top sites and longtail to see if there is a strong difference among the various categories, each of which correspond to different tiers of cost.

8.1 CA Popularity and Prevalence

We start by identifying which certificate authorities are most prevalent among the domains in our data set. For certificate data, we use the snapshots of crawls of the Alexa Top Million domains by Censys [18]. Note that the Alexa snapshot used by Censys is from a slightly different time than the one we used for crawling with `pshtt`, which leads to slightly different domain sets; in the Censys snapshot, 4,824 top sites (48%) had certificates and 368,250 (37%) longtail sites had certificates.

Table 4 shows the top ten certificate issuers by issuer organization for both the top and longtail sites. Not only does Let's Encrypt issue far more certificates among longtail sites (17%) than top sites (4%), it is the second most prevalent issuer for longtail sites. As a free service, Let's Encrypt clearly has attraction for longtail sites, suggesting that certificate cost is a notable factor for them. (We performed the same analysis using the issuer common name field from the certificates, with nearly identical results in prevalence and ranking for Let's Encrypt between the two kinds of sites.)

Top Sites		Longtail	
Certificate Authority	% of Tail	Certificate Authority	% of Tail
Comodo	26.3%	Comodo	26.3%
GeoTrust Inc.	9.8%	Let's Encrypt	16.9%
Symantec	7.6%	GeoTrust Inc.	8.1%
GlobalSign nv-sa	6.1%	GoDaddy.com, Inc.	5.6%
DigiCert Inc	6.0%	cPanel, Inc.	5.0%
GoDaddy.com, Inc.	4.6%	GlobalSign nv-sa	4.0%
Let's Encrypt	4.0%	DigiCert Inc	2.9%
Amazon	3.0%	Symantec	2.7%
thawte, Inc.	2.6%	thawte, Inc.	2.0%
Google Inc	2.5%	Google Inc	1.9%

Table 4: Percentage of top and longtail sites with valid HTTPS categorized by their certificate authorities.

8.2 Migration between CAs

As another indicator of cost sensitivity, we examine whether cost is a factor when sites migrate from one certificate authority to another. A site that already has a certificate presumably is already familiar with the existing CA; moving to a different CA introduces a new process and new overhead, which suggests other factors for switching CAs, such as cost. As such, we focus on just whether sites transfer from a CA that charges for certificates to the free Let's Encrypt service.

To determine whether a site changed CA, and the initial and final CAs used by sites that changed, we use Censys certificate snapshots over time. For sites in our data with certificates on August 31, 2017, we check monthly snapshots back in time to August 2015. If the certificate issuer organization changed in this period, we label the site as transferred. If it never changed, we label it as new. If a site had at least one month without any CA (e.g., perhaps if a site changed ownership), then we label the site as new using the new CA.

Focusing on the CAs that are major providers of certificates, sites in the longtail are far more likely to transfer to the free service Let's Encrypt than top sites. Starting with the top 20 CAs by number of sites for the top sites, we intersect them with the top 20 CAs by the same metric for the longtail.

Table 5 shows the 12 CAs in this intersection, which together contain 25% of total sites in the top and 16.8% of total sites in the longtail. Across the CAs, the percent of sites that transfer to Let's Encrypt in the longtail is substantially larger (4×) than the percent of sites that transfer to Let's Encrypt in the top sites. Taking all of the sites in these 12 CAs as a whole, over 28.6% of CA transfers in the longtail are to Let's Encrypt while only 7.2% of the same CAs see top sites transfer to Let's Encrypt. When considering the total over *all* CA transfers, the results are the same: across all CAs, 7.7% of the top sites transfer to Let's Encrypt and 29% of longtail sites. Moreover, when examining these 12 CAs, over 4× as many sites in the longtail and over 2× in the top sites transfer to Let's Encrypt than from Let's Encrypt to another CA.

8.3 Types of Certificate Validation

Finally, we consider how prevalent the types of certificate validations are among top and longtail sites. CAs can promote three different types of domain validation. Some organizations like Let's Encrypt only focus on domain validation (DV): if you can prove you own

Certificate Authority	Top Sites	Longtail
Comodo	11.3%	35.9%
GlobalSign	2.9%	29.4%
GeoTrust	5.7%	23.1%
Symantec	2.1%	4.3%
DigiCert	5.5%	29.0%
GoDaddy	5.6%	19.0%
Thawte	5.1%	22.1%
StartCom	32.4%	51.7%
cPanel	5.6%	36.4%
Starfield	9.1%	12.7%
WoSign	22.6%	42.4%
Gandi	27.5%	69.9%
Top Combined	7.6%	28.6%
All CAs	7.7%	29%

Table 5: Percentage of sites that transfer CAs to Let's Encrypt among the major CAs for both top and longtail sites. Longtail sites are much more likely to transfer to Let's Encrypt.

the domain, you can get a certificate. Other CAs also offer organization validation (OV): it has a more manual validation protocol that some CAs purport prevent fake websites. Finally, there is extended validation (EV): in addition to having an even more manual validation protocol for domains than OV, some browsers will also show a different visual representation. Again, some CAs advertise EV as more secure than both OV and DV [29]. Typically, DV is cheaper than OV, and both are cheaper than EV. For example, for a single domain GoDaddy's pricing is $60 for a DV certificate, $103 for an OV certificate, and $199 for an EV certificate [28]. Costs are correspondingly higher for a certificate for multiple domains, or a wildcard certificate.

Comparing the proportions of DV, OV, and EV between top and long sites, we find that top sites pay for the more expensive validations more often. In the longtail, 63.9% of certificates are DV, while just 12.3% are OV and 4.2% are EV (with the rest being unknown). Among top sites, though, significantly more use the more expensive validations: only 42.5% are DV, while 28.6% are OV and 7.6% are EV. Once again, longtail sites appear to be more sensitive to cost than top sites.

8.4 Discussion

Across all perspectives, certificate cost does appear to be a significant factor for longtail sites: Let's Encrypt is 4× more prevalent among longtail than top sites; when sites transfer CAs, longtail sites are 3.7× more likely to transfer from a paid CA to Let's Encrypt; and top sites are more likely to use premium certificate validation services than longtail sites.

9 CERTIFICATE PERSISTENCE

More generally, HTTPS adoption includes sites supporting HTTPS for the first time, as well as sites that continue to use it over time. If sites that have HTTPS do not persist, then adoption will increase at a slower rate, or even stagnate. Until now, we have only considered sites supporting HTTPS for the first time. As a final analysis, we examine the behavior of sites continuing to support HTTPS over time. We study the behavior of sites renewing their certificates, or certificate persistence, as another aspect of HTTPS adoption.

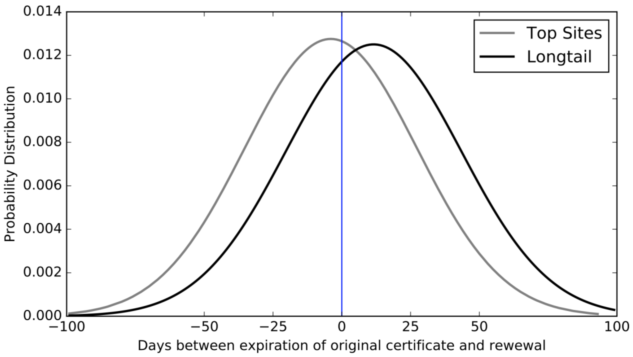

Figure 5: PDFs of the time between expiration and renewal for both the top and longtail sites, with a vertical line at 0 for reference. Top sites are more likely to renew their certificate before expiration while longtail sites are more likely to renew after.

We focus on sites that were valid on August 31, 2017 and December 17, 2017 in the Censys certificate snapshots, and had certificates that expired between the two dates and were eventually renewed. A similar fraction of sites expired in this period, with 11% (1097) of top and 10% (99081) of longtail sites expiring. We use this subset of sites moving forward.

For sites in both datasets, we see a positive trend for certificate persistence. Encouragingly, only a tiny fraction of both types of sites do not renew: just 0.05% of the top sites and 0.08% of the longtail sites. During this period, we also see 11% of top sites and 4.9% of longtail sites acquire a valid certificate for the first time, which shows that top sites are still adopting HTTPS at faster rates. Once sites have HTTPS, they are likely to maintain it, regardless of whether they are a top or longtail site.

While certificate persistence is similar among top and longtail sites, we also look at the certificate renewal behavior more closely. For sites that renew their certificate, we determine how long it took them to renew by computing the difference between the end date of the certificate they had in August and the begin date of the new certificate they had in December. Sites with a negative difference renew their certificates before expiration, while sites with a positive difference let their certificate expire before renewing.

Overall, top sites are more likely to be proactive in their certificate renewal, while longtail sites are more reactive. Figure 5 shows the time between expiration and renewal for both top and longtail sites as PDFs. For clarity we only show renewal periods between -100 and 100 days, which excludes just 2% of top and 0.7% of longtail sites. Top sites are more likely to renew their certificate before expiration (59% of top sites do so), while longtail sites are more likely to renew their certificate after expiration (60% of longtail sites do so). The average time for a top site to renew its certificate was -9.4 days, with a median of -7 days. For longtail sites, the average renewal time was 9 days, with a median of 15. While persistence is the same, the underlying renewal behavior differs between top and longtail sites.

Discussion. Overall, sites in both the top and longtail renew existing certificates, which bodes well for HTTPS adoption: improving adoption rates can focus solely on sites that have never supported HTTPS. Sites already supporting HTTPS likely have a familiar mechanism for renewal, and it is also possible that sites have an incentive to keep HTTPS since their visitors and customers are expecting it.

10 CONCLUSION

Our goal has been to understand, at scale, whether there are sectors of the web site population, particularly the less popular sites, that might be more amenable to upgrading their security posture by adopting HTTPS. By identifying correlates of HTTPS deployment, we hoped to identify shared infrastructure, motivations or constraints to drive further adoption. In our analyses we found a number of negative results, but also some positive patterns.

For instance, while we had hoped that newer sites would be more likely to adopt HTTPS given increased awareness and support for security features (and hence the HTTPS deployment problem would be self-correcting), we found no strong correlation between the age of sites and whether they support HTTPS. Indeed, large fractions of even recent new sites do not support HTTPS.

More positively, two factors that do correlate with HTTPS adoption are transparent hosting support and cost. Services such as Cloudflare, Google, and OVH offer both free certificates as well as automatically enabling HTTPS without the need for any customer action. These features provide a clearly compelling combination for increasing HTTPS adoption. Among sites using hosting providers, Cloudflare in particular is both the most popular provider and has the highest adoption of HTTPS (77% for top sites on Cloudflare, and 81% for longtail sites). Motivating more hosting providers to provide free certificates and, in particular, automatically enabled HTTPS support could have a significant impact on HTTPS adoption.

Further underscoring the importance of cost, the expense of certificates also appears to be a significant factor on its own, particularly for longtail sites where the free service Let's Encrypt is 4× more likely than among top sites. Further, when sites transfer CAs, longtail sites are 3.7× more likely to transfer from a paid CA to Let's Encrypt. Offering free certificates has clearly had a positive impact on HTTPS adoption.

REFERENCES

[1] Adrian, D., Bhargavan, K., Durumeric, Z., Gaudry, P., Green, M., Halderman, J. A., Heninger, N., Springall, D., Thomé, E., Valuenta, L., VanderSloot, B., Wustrow, E., Zanella-Béguelin, S., and Zimmermann, P. Imperfect Forward Secrecy: How Diffie-Hellman Fails in Practice. In *Proceedings of the 22nd ACM Conference on Computer and Communications Security (CCS)* (Oct. 2015).

[2] Akhawe, D., Amann, B., Vallentin, M., and Sommer, R. Here's My Cert, So Trust Me, Maybe?: Understanding TLS Errors on the Web. In *Proceedings of the 22nd International World Wide Web Conference (WWW)* (May 2013), pp. 59–70.

[3] Akhawe, D., and Felt, A. P. Alice in Warningland: A Large-Scale Field Study of Browser Security Warning Effectiveness. In *Proceedings of the 22nd USENIX Security Symposium* (Washington, D.C., Aug. 2013), pp. 257–272.

[4] Alexa. Alexa Top 1 Million Download. http://s3.amazonaws.com/alexa-static/top-1m.csv.zip. Accessed: 2018-05-24.

[5] Amazon. Cloudfront Pricing. https://aws.amazon.com/cloudfront/pricing/. accessed: 2018-11-04.

[6] Amazon. Elastic Beanstalk. https://aws.amazon.com/elasticbeanstalk/. accessed: 2018-11-04.

[7] Amazon. Elastic Compute Cloud. https://aws.amazon.com/ec2/. accessed: 2018-11-04.

[8] Amazon. Lightsail. https://aws.amazon.com/lightsail/. accessed: 2018-11-04.

[9] Amazon. Simple Cloud Storage Service. https://aws.amazon.com/s3/. accessed: 2018-11-04.

[10] Amazon. Wordpress. https://aws.amazon.com/getting-started/tutorials/launch-a-wordpress-website/. accessed: 2018-11-04.

[11] Aviram, N., Scinzel, S., Somorovsky, J., Heninger, N., Dankel, M., Steube, J., Valenta, L., Adrian, D., Halderman, J. A., Dukhovni, V., Käsper, E., Cohney, S., Engels, S., Paar, C., and Shavitt, Y. DROWN: Breaking TLS with SSLv2. In *25th USENIX Security Symposium* (Aug. 2016).

[12] Basques, K. Why HTTPS Matters. https://developers.google.com/web/fundamentals/security/encrypt-in-transit/why-https. Accessed: 2018-11-04.

[13] Caddy. https://caddyserver.com/. accessed: 2018-10-28.

[14] Cangialosi, F., Chung, T., Choffnes, D., Levin, D., Maggs, B. M., Mislove, A., and Wilson, C. Measurement and Analysis of Private Key Sharing in the HTTPS Ecosystem. In *Proceedings of the 2016 ACM Conference on Computer and Communications Security (CCS)* (Vienna, Austria, Oct. 2016), pp. 628–640.

[15] Chung, T., Liu, Y., Choffnes, D., Levin, D., Maggs, B. M., Mislove, A., and Wilson, C. Measuring and Applying Invalid SSL Certificates: The Silent Majority. In *Proceedings of the 2016 Internet Measurement Conference (IMC)* (Santa Monica, CA, USA, Nov. 2016), pp. 527–541.

[16] Cloudflare. https://www.cloudflare.com/. accessed: 2018-11-04.

[17] DigitalOcean. https://www.digitalocean.com/. accessed: 2018-11-04.

[18] Durumeric, Z., Adrian, D., Mirian, A., Bailey, M., and Halderman, J. A. A Search Engine Backed by Internet-Wide Scanning. In *Proceedings of the 22nd ACM Conference on Computer and Communications Security (CCS)* (Oct. 2015), pp. 542–553.

[19] Durumeric, Z., Kasten, J., Bailey, M., and Halderman, J. A. Analysis of the HTTPS Certificate Ecosystem. In *Proceedings of the 2013 Internet Measurement Conference* (New York, NY, USA, 2013), IMC '13, ACM, pp. 291–304.

[20] Durumeric, Z., Li, F., Kasten, J., Amann, J., Beekman, J., Payer, M., Weaver, N., Adrian, D., Paxson, V., Bailey, M., and Halderman, J. A. The Matter of Heartbleed. In *Proceedings of the 2014 Internet Measurement Conference* (New York, NY, USA, 2014), IMC '14, ACM, pp. 475–488.

[21] EFF. Certbot. https://certbot.eff.org/. accessed: 2018-11-04.

[22] Felt, A. P., Ainslie, A., Reeder, R. W., Consolvo, S., Thyagaraja, S., Bettes, A., Harris, H., and Grimes, J. Improving SSL Warnings: Comprehension and Adherence. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (New York, NY, USA, 2015), CHI '15, ACM, pp. 2893–2902.

[23] Felt, A. P., Barnes, R., King, A., Palmer, C., Bentzel, C., and Tabriz, P. Measuring HTTPS Adoption on the Web. In *26th USENIX Security Symposium (USENIX Security 17)* (Vancouver, BC, 2017), USENIX Association, pp. 1323–1338.

[24] Felt, A. P., Reeder, R. W., Ainslie, A., Harris, H., Walker, M., Thompson, C., Acer, M. E., Morant, E., and Consolvo, S. Rethinking Connection Security Indicators. In *Twelfth Symposium on Usable Privacy and Security (SOUPS 2016)* (Denver, CO, 2016), USENIX Association, pp. 1–14.

[25] Freedom of the Press Foundation. Secure the News. https://securethe.news/. Accessed: 2018-11-04.

[26] GoDaddy. https://www.godaddy.com/. accessed: 2018-11-04.

[27] GoDaddy. GoDaddy Forums on Let's Encrypt. https://www.godaddy.com/community/SSL-And-Security/SSL-Certificates-Paid-vs-Free/td-p/729. accessed: 2018-11-04.

[28] GoDaddy. GoDaddy SSL Cost. https://www.godaddy.com/web-security/ssl-certificate. accessed: 2018-11-04.

[29] GoDaddy. Secure your data and boost your Google rank. https://www.godaddy.com/web-security/ssl-certificate. Accessed: 2018-11-04.

[30] GoDaddy. Website Builder. https://www.godaddy.com/websites/website-builder. accessed: 2018-11-04.

[31] GoDaddy. Website Hosting. https://www.godaddy.com/hosting/web-hosting. accessed: 2018-11-04.

[32] GoDaddy. Wordpress Hosting. https://www.godaddy.com/hosting/wordpress-hosting. accessed: 2018-11-04.

[33] Google. App Engine. https://cloud.google.com/appengine/. accessed: 2018-11-04.

[34] Google. Blogger. https://www.blogger.com. accessed: 2018-11-04.

[35] Getting The Green Lock: HTTPS Stories from the Field. https://www.youtube.com/watch?v=GoXgl9r0Kjk. Accessed: 2018-11-04.

[36] Google. Google Sites. https://sites.google.com/. accessed: 2018-11-04.

[37] Google. Google Support for HTTPS. https://support.google.com/webmasters/answer/6073543. Accessed: 2018-11-04.

[38] Google. Google Transparency Report Overview. https://transparencyreport.google.com/https/overview. Accessed: 2018-11-04.

[39] Google. Google Transparency Report Top Sites. https://transparencyreport.google.com/https/top-sites. Accessed: 2018-11-04.

[40] Government, U. S. Pulse. https://pulse.cio.gov/. Accessed: 2018-11-04.

[41] Hetzner. https://www.hetzner.com. accessed: 2018-11-04.

[42] Holz, R., Braun, L., Kammenhuber, N., and Carle, G. The SSL Landscape: A Thorough Analysis of the x.509 PKI Using Active and Passive Measurements. In *Proceedings of the 2011 ACM Internet Measurement Conference* (2011),

IMC '11, ACM, pp. 427–444.

[43] Internet Archive. https://www.archive.org/. accessed: 2018-11-04.

[44] Python ipwhois package. https://pypi.org/project/ipwhois/. accessed: 2018-11-04.

[45] KROMBHOLZ, K., MAYER, W., SCHMIEDECKER, M., AND WEIPPL, E. "I Have No Idea What I'm Doing" — On the Usability of Deploying HTTPS. In *26th USENIX Security Symposium (USENIX Security 17)* (Vancouver, BC, 2017), USENIX Association, pp. 1339–1356.

[46] KUMAR, D., MA, Z., DURUMERIC, Z., MIRIAN, A., MASON, J., HALDERMAN, J. A., AND BAILEY, M. Security Challenges in an Increasingly Tangled Web. In *Proceedings of the 26th International World Wide Web Conference* (Republic and Canton of Geneva, Switzerland, 2017), WWW '17, International World Wide Web Conferences Steering Committee, pp. 677–684.

[47] LIANG, J., JIANG, J., DUAN, H., LI, K., WAN, T., AND WU, J. When HTTPS Meets CDN: A Case of Authentication in Delegated Service. In *Proceedings of the 2014 IEEE Symposium on Security and Privacy* (Washington, DC, USA, 2014), SP '14, IEEE Computer Society, pp. 67–82.

[48] Linode. https://www.linode.com/. accessed: 2018-11-04.

[49] LIU, Y., TOME, W., ZHANG, L., CHOFFNES, D., LEVIN, D., MAGGS, B., MISLOVE, A., SCHULMAN, A., AND WILSON, C. An End-to-End Measurement of Certificate Revocation in the Web's PKI. In *Proceedings of the 2015 Internet Measurement Conference* (2015), IMC '15, ACM, pp. 183–196.

[50] OVH. https://www.ovhtelecom.fr/. accessed: 2018-11-04.

[51] Pushing HTTPS. https://github.com/dhs-ncats/pshtt. Accessed: 2018-11-04.

[52] WILLIAMS, C. Let's Encrypt Won its Comodo Trademark Battle — but now fan tools must rename. https://www.theregister.co.uk/2016/09/18/letsencrypt_trademark_clash/, Sept. 2016. accessed: 2018-11-04.

[53] ZDNS. https://github.com/zmap/zdns. accessed: 2018-11-04.

[54] ZHANG, L., CHOFFNES, D., LEVIN, D., DUMITRAŞ, T., MISLOVE, A., SCHULMAN, A., AND WILSON, C. Analysis of SSL Certificate Reissues and Revocations in the Wake of Heartbleed. In *Proceedings of the 2014 Internet Measurement Conference* (New York, NY, USA, 2014), IMC '14, ACM, pp. 489–502.