How to bring HTTPS to the masses?
Measuring issuance in the first year of Let's Encrypt

Maarten Aertsen
maarten@rtsn.nl rsa4096/0x058B121814789500

December 1, 2016

Master of Science thesis

Committee:	Maciej Korczyński,	(Delft University of Technology)
	Jan van den Berg,	(Delft University of Technology, Leiden University)
	Giovane C.M. Moura	(SIDN Labs)
Programme:	Exec.MSc Cyber Security	
Institution:	Leiden University, The Netherlands	

Abstract

The World Wide Web is the most popular application of the Internet. Pervasive monitoring affects its users by compromising the confidentiality of their communications. Even where the encryption technology is available to mitigate pervasive monitoring, technical complexity and adverse economic incentives have delayed its widespread deployment. As a result, the bulk of the Web and its users remain powerless against this threat. The question is: *how to bring HTTPS to the masses*? In a case study, we examine efforts by the certificate authority *Let's Encrypt* to address both technical complexity and financial barriers to widespread deployment. We evaluate who has been using *Let's Encrypt* in the first year since its inception. To gauge its contribution to the democratization of encryption technology on the Web, we measure growth in domain coverage, adoption by popular and large versus small players, the type of adopters and their perseverance. We find a lower bound of 2% global coverage on the monthly use of *Let's Encrypt* aggregated at second level domains. Dominant driver of this growth are hosting companies (68%), in particular those bulk certifying domains of their users (3 companies cover 47%). It is exactly these companies, serving numerous, smaller customers that would otherwise not enable the use of encryption by their visitors. The approach pursued by *Let's Encrypt* in its first year of operation is worthy of broader consideration and adoption in the industry. Though issuing for free will not fit the majority of business models, the adoption of its advances in automation may prove pivotal to bring HTTPS to the masses. For security problems more generally, taking a market approach to deploy technical mitigations may be well worth future consideration.

Contents

Chapter 1

Introduction

The World Wide Web (hereafter Web) is the most popular application of the Internet. It facilitates interaction between people, greatly decreasing opportunity cost for communication and transaction. Whether it is online shopping; reading the news; paying taxes or voicing your opinion on a social network, the Web facilitates interaction that is so convenient that societies have come to depend upon it.

Content on the Web is made accessible online by those that host websites. The overwhelming majority of end-users, *the masses*, do not self-host content. Web hosting is a specialized service: running personal or (small) business websites is dominantly delegated to providers.

1.1 Problem description

The disclosures in the wake of Edward Snowden have shown the reality of pervasive monitoring on the Internet. Pervasive monitoring affects the activities of its users by compromising the confidentiality of their communications. Though essentially *"a technical attack that should be mitigated in the design of [..] protocols, where possible"* (IETF, [11]), getting mitigations deployed is a difficult problem on its own.

Software to encrypt client-server communication on the Web has been available as early as 1994, evolving to what we now know as the Hypertext Transfer Protocol Secure (HTTPS) protocol. Yet deployment has lagged and not without reason. Until recent years, organisations have had little incentive to deploy [21]. But when it comes to use of encryption by the masses, hosting providers have a dominant impact on whether or not an end-user is able to communicate with preservation of confidentiality. Granted, some of the most popular services (Facebook, Apple iCloud, Google) have spear-headed deployment within their ecosystem representing a sizeable timeshare of Internet use [2]. But continued progress eventually requires uptake by websites that are not self-run, but hosted. As a result, the bulk of the Web and its users remain powerless against the monitoring threat.

1.2 Research questions

How to bring HTTPS to the masses? A question the size of an elephant. Perhaps an excellent topic for a dissertation and most certainly a question that has captured the author. Here, we take our first small bite by performing a case study. In this study we examine efforts by a market entity, the certificate authority *Let's Encrypt*, that addresses both technical complexity and financial barriers to widespread deployment of HTTPS.

Main question

> *"How to bring HTTPS to the masses?"*

Subquestions

1. *"What prevents widespread adoption of HTTPS?"*

   This question should result in a summary of reasons for delayed adoption, covering incentives and barriers to deployment. Knowing that the masses do not self-host, special consideration should be given to the market segment of hosting.

2. *"How does Let's Encrypt contribute to widespread adoption?"*

   This question aims to make visible the approach taken by *Let's Encrypt* to counteract the factors delaying adoption.

3. *"Who has been using Let's Encrypt in the first year since its inception?"*

   This question aims to measure actual use of the service since its adoption, thereby giving a lower bound on its potential.

4. *"What insights do the results from the case study on Let's Encrypt provide for bringing HTTPS to the masses?"*

   The final question brings together our case study on *Let's Encrypt* and the main question, with the aim of making accessible our observed results for the larger stated goal that inspired the title of this thesis.

1.3 Approach

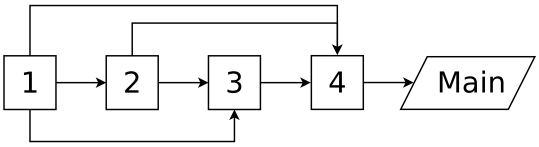


Figure 1: Research flow

The flow of research is as depicted in Figure 1. item 1 is answered directly based on the current body of knowledge. item 2 depends on both the answer to item 1 and the body of knowledge. Answering the first two questions hence involves literature review and where results are yet unpublished by investigating other public sources.

The answer to item 2 in turn enables us to hypothesize *Let's Encrypt*'s potential to contribute to widespread adoption. In turn, this estimated potential feeds into the design of a measurement study to answer item 3.

item 3 involves empiral research and results in both raw results and specific observations to answer stated question. Combining the observations answering item 3 with the answers to item 1 and item 2 yield insights that answer item 4. These in term help us to move one step closer to answering our main question. The next section describes the flow of this thesis in answering these respective questions in turn.

1.4 Outline

This thesis is structured as follows. Continuing after this introduction Chapter, chapter 2 fills in the background of this research before discussing incentives and costs delaying HTTPS adoption (addressing item 1), *Let's Encrypt*'s contribution (addressing item 2) as well as related work. chapter 3 treats research design, data sets and methodology. Then chapter 4 covers the results of our emperical study (addressing item 3), which are subsequently discussed in chapter 5, where future work is also covered. Finally, main results and conclusions are subject of chapter 6.

Chapter 2

Background

This chapter is based on *"Barriers to HTTPS adoption in the shared web hosting segment"* by M. Aertsen.

This Chapter provides the background on the research that is subject of this thesis. It gives context to the problem statement and explains the particular focus on *Let's Encrypt*. With respect to context, we start with a description of the activities impacted by the problem statement, then describe a traditional governance angle. We proceed with alternative regulatory approaches, introducing HTTPS deployment seen from a market angle. We look at the reasons for delayed deployment, covering incentives and costs. Then *Let's Encrypt* is introduced and we argue what makes it worthy of attention. Finally, we discuss related research. After reading this Chapter, the reader should understand what makes *Let's Encrypt* interesting in attempting to answer the question "How to bring HTTPS to the masses?".

2.1 Societal relevance

The Internet facilitates interaction between people, greatly decreasing opportunity cost for communication and transaction. Whether it is online shopping; reading the news; paying taxes or voicing your opinion on a social network, the Web facilitates interaction that is so convenient that societies have come to depend upon it. Every successful use builds trust in its ability to service their users' needs. Communication over the Web requires such trust. Trust among communicating peers, trust in the ability to express oneself but especially trust in the ability of the Web to convey in a way much like offline communication.

Such trust in the Web's ability is not always warranted. Between commercial interests of companies [34] and dragnet legislation by governments in the name of public safety [28] there is increasing pressure on the ability to browse the Web without snooping bystanders. That there is a need for privacy in social interaction has been broadly established [37]: snooping and tracking present a threat to the activity of using the Web to interact.

Now, threats to privacy are hardly new. In each wave of technological development, be it state mail or telegraph institutions [31, p.11] or the introduction of the Kodak [36], Governments have seen a need to protect their citizen's ability to have private interaction. Such protection has been extended to the Internet. Still, regulation does nothing to bolster its citizen's resilience. With a global Internet and differing opinions on the role of the state to curate its contents, there is value in inherent protection in addition to that provided by the law.

Software has been available as early as 1994 to encrypt client-server communication on the Web, evolving to what we now know as the HTTPS protocol. Encryption technology enables its users to shield their interaction not from discovery or observation, but from access to its contents. Asghari et al. suggest wide consensus on the belief "that the average end-user cannot reasonably be expected to exert control over the HTTPS ecosystem" [5]. The overwhelming majority of end-users, "the masses", do not self-host content on the Web. Web hosting is a specialized service: running personal or (small) business websites is dominantly delegated to providers. And when it comes to use of encryption, the hosting provider has equal or greater impact on whether or not an end-user is able to communicate with preservation of confidentiality.

Yet deployment has lagged and not without reason. Until recent years, organizations have had little incentive to deploy [21]. This appears to be slowly shifting, with more (commercial) visibility on such topics as advertisement injection [33], HTTPS dependent functionality [22] and the introduction of search engine optimization (SEO) incentives [6]. And while several of the largest Web properties have long started conversion resulting in noticeable uptake [2], continued progress eventually requires uptake by websites that are not self-run, but hosted. These are the market segments of (shared) hosting, serving both individuals and smaller organizations, where HTTPS needs to make economic sense for the hosting provider to consider adoption.

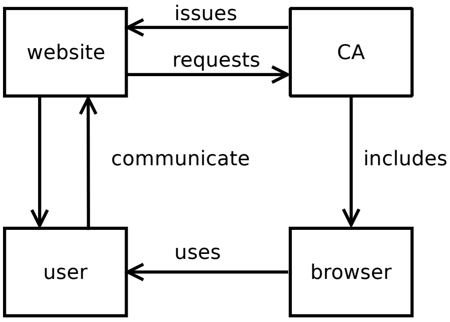


Figure 2: Entities in HTTPS: user behind browser, website and CA

2.2 Concepts

In this subsection, we introduce various concepts relevant to the research questions at hand. In order we treat HTTPS, Domain names and Web hosting.

HTTPS HTTPS is the composition of HTTP, the communications protocol underpinning the Web, and SSL/TLS–a protocol providing encryption capabilities (hereafter TLS). If one views HTTP as the means for a customer on the Web to talk to a shop owner, then TLS serves to prevent random bystanders from switching the price tag prior to check-out or from inspecting your cart while browsing the store. There are three main entities involved with a TLS connection (Figure 2): a client, say the user behind his or her browser; the server, e.g. running web shop software for a small company and the Certificate Authority (CA), vouching for the online identity of the web shop. The CA is not involved in the actual connection, but plays a role in the trust relation underpinning the TLS communication. The CA is known to the users' client, which therefore is able to validate the shop's *certificate*, an assertion (issued) by the CA that ties the encryption key presented to the identity of the shop. To set up a website for HTTPS, an X.509 certificate needs to be obtained (requested) from a CA and appropriate server software needs to be configured[1]. Certificate issuance has traditionally been a manual process of submitting a request (CSR) for validation and signing by a CA.

Domain names Domain names are used to provide a simple identification label for hosts, services, applications, and networks on the Internet [27]. When a domain name is qualified to such extent that it can serve as a label, it is referred to as a Fully Qualified Domain Name (FQDN). For example: `example.org` is a 2^{nd}-level domain, which may contain the FQDN `www.example.org`. We define *domain* as 2^{nd}–level or 3^{rd}–level if a given TLD (Top Level Domain) registry provides such registrations, e.g. `example.uk`, `example.co.uk`, etc.

Web hosting Web hosting is the industry that maintains content online on the Web on behalf of customers, paying directly or indirectly, e.g. through the addition of ads. Shared web hosting is when multiple websites each identified by their own domain name are hosted on one server, sharing a single IP address. More formally, following prior research by Tajalizadehkhoob et al., shared web hosting is defined as hosting more than 10 distinct domains[2] on a single IP address [32]. (Shared) hosting specifically represents the challenge of dealing with a large number of domains each owned by their own customer. It becomes apparent that for a shared hosting provider to consider HTTPS as a service for its customers, scale matters a lot.

2.3 The deployment of HTTPS

The current state of HTTPS deployment in the world would suggest that adoption is not appealing enough to convince every shared hosting provider [2]. In the next two subsections we discuss reasons for delayed deployment. First, we cover incentives: the

[1] Proper configuration of SSL/TLS protocols (HTTPS included) in the presence of a valid certificate is a continuous challenge in itself and is not the topic of this work.

[2] Distinct second level domains are counted in order to exclude hosts with 10 or more Fully Qualified Domain Names (hereafter FQDNs) primarily differing in the lower level parts of the same second level domain. E.g. {suchduplicate, verysame, manydouble, wow} .example.org, etc. all pointing to the same IP address.

positive and negative stimuli for players involved. Then we cover inherent cost factors associated with deployment which may cause a player with insufficient incentives not to proceed. The next section then covers *Let's Encrypt* and what makes it worthy of attention with the context of this section in mind. The attentive reader may note that costs can be equated with negative incentives, but we have chosen to call them out separately to emphasize their particular influence irrespective of other incentives in play.

2.3.1 Incentives

We are able to distinguish three distinct positive incentives to deploy HTTPS in shared web hosting, ordered by decreasing strength in the mind of the author. First, deploying HTTPS can be a distinguishing factor[3] in a competitive market. Such investment may be part of branding on quality, security or perhaps timeliness for support of new web standards. Second, a hosting provider may want to implement HTTPS in response to user demand, or to attract users with use-cases that require HTTPS (e.g. web shops) that would not normally be interested in shared web hosting. Finally, HTTPS may one day become part of any standard web hosting offering, much like support for file uploads or e-mail has become. A hosting provider not providing the standard offering may lose customers comparing the few differences in a commodity market.

There are also negative incentives at play. The most convincing negative incentive would be to reserve HTTPS as a premium feature reserved for more profitable hosting packages. One could imagine a hosting provider selling web hosting employing differentiated pricing and offering it as part of dedicated (i.e. non-shared) hosting only. Finally, we note that where positive incentives are not strong enough, it will be the opportunity cost that determines the investment pattern of hosting providers. Having provided an overview of positive and negative incentives, we proceed to describe costs facing hosting providers deciding to deploy HTTPS on their properties.

2.3.2 Costs

Shared hosting is an example of a service to the public that has traditionally not seen widely employed encryption. Kasten suggests that there are two main factors hindering HTTPS deployment[21, p.125]: certificate cost and deployment time. We will deepen our understanding of both and add a third category: the cost of additional complexity. For ease of reference the different categories are split out below.

```
1 certificate cost
  - monetary cost of purchase
```

[3] Though investment in security is generally hard to observe (an example of asymmetric information), hosting providers who decide to roll out HTTPS give their tenants visible return on investment due to the (padlock) signalling built into browsers.

```
2 deployment cost
  - initial deployment
  - certificate renewal
3 complexity cost
  - familiarity cost
  - cost of forgotten renewal
```

For sake of completeness we will also mention two non cost factors which have held back HTTPS for years but no longer offer serious resistance. TLS was long considered to be slow or resource intensive. This has been thoroughly debunked in recent years, both because implementations have grown more mature, but also because computing power has increased and expensive operations have been embedded in silicon[14]. For shared hosting specifically, there was always the issue of SNI. SNI or “Server Name Indication” is an extension to TLS to support serving different certificates for different FQDNs from a single IP address (notably the specific scenario of shared web hosting). Windows XP and its default browser Internet Explorer have never had support for SNI. But with support for Windows XP being discontinued in April 2014, its usage numbers have been dwindling, approaching 0.5% globally at the time of writing. Both may well have contributed to lagging adoption, but should no longer be part of current considerations.

We continue with costs that are still current. First and most straightforward, certificate cost. This is the monetary cost of certificates. Asghari et al. have shown the large price differences among certificates that are essentially the same product minus some value added services [5].

Second, deployment time can be split into (initial) deployment and repeated deployment. This is due the fact that certificates have a limited validity period, usually one or two years. As a result the cost of the request and configuration process is a recurring one.

Third, deploying HTTPS introduces a complexity cost. Primarily because staff will need to become familiar with the intricacies of (repeated) certificate request, reconfiguration and troubleshooting. Complexity may also lead to failure: there is a very real cost to deploying HTTPS and then forgetting renewal as this leads to service downtime (and scary warning screens for end-users). With incentives and costs discussed, it is time to examine the approach take by *Let’s Encrypt* .

2.4 Let’s Encrypt’s contribution

Let’s Encrypt is the first Internet deployed robot CA. Contrary to most contemporary CAs, there is no web form based process for certificate issuance. Instead, the CA speaks to client software using a protocol (ACME), due to be standardized[7]. ACME client software can be run autonomously after accepting the terms of service. In the period since its public release in November 2015, multiple ACME client implementations have been created, developed in parallel for various use cases.

Let's Encrypt is run by the non-profit ISRG (Internet Security Research Group, a California public benefit corporation). ISRG [17] has the stated mission to *"to reduce financial, technological, and education barriers to secure communication over the Internet"*. In particular, ISRG's CA, *Let's Encrypt* , aims to increase TLS (and thereby HTTPS) adoption on the Internet by making the process of obtaining certificates free, automatic, secure, transparent, open and cooperative [18, 7].

Let's Encrypt does not charge for certificate issuance nor revocation thereof. We note that free certificates have always existed from competing CAs such as StartCom, though they charged for revocation which is not the case with *Let's Encrypt* . The *Let's Encrypt* business model does not revolve around getting paid for certification. Instead, the resources required to operate are donated by a set of sponsors/partners, including a number of significant players normally paying for their certificates. The validation required from a CA prior to issuing a certificate for a domain is also automated. Notably, *Let's Encrypt* only issues Domain Validated (DV) certificates where validation requires assessing control over the domain without interaction with their human owners. This brings down marginal cost for validation per certificate to near zero, in line with other information products. Having set up the required infrastructure and assuming production within maximum capacity, operational expenses are thus dominated by the fairly fixed cost of keeping the robot CA running [4].

```
1 certificate cost
  - monetary cost of purchase
2 deployment cost
  - initial deployment
  - certificate renewal
3 complexity cost
  - familiarity cost
  - cost of forgotten renewal
```

We will now examine *Let's Encrypt* 's contribution based on the costs described in the previous section, reproduced for ease of reference above. Certificate purchase cost is zero and unlikely to raise any time soon given ISRG's mission. This is a monetary differentiator against a part of the market, representing the removal of a significant cost. Deployment cost naturally remains. However, with the repeated certificate request, issuance and installation replaced by a one-time installation/configuration of an ACME client, the certificate renewal cost is cut[4]. This is a differentiator versus the full CA market, representing the removal of a significant recurring expense. In the third category of complexity cost, *Let's Encrypt* , by virtue of its support for automating renewal by means of ACME clients reduces the chance of service downtime due to forgotten certificate renewal. Finally, one may argue that some ACME clients are especially tailored

[4] ACME clients being software, there is a recurring cost of updating and patching, yet these costs are amortized over all domains.

to reduce human configuration effort. We counter that the typical shared hosting environment may be harder to automatically configure than the stock configurations they target, which is why we don't list this as a significant differentiator.

```
1 (gone)
2 deployment cost
  - initial deployment
  - (replaced by updating/patching)
3 complexity cost
  - familiarity cost
  - (gone)
```

All in all, this yields a reduced list of costs for *Let's Encrypt* (shown above), which the authors belief represents its primary contribution to lower barriers to HTTPS adoption.

2.4.1 Relevance to the masses

We will now argue why we believe the contribution described may be especially significant for the democratization of HTTPS. As stated, the mass of users do not self-host, but make use of the services of the (shared) hosting market. Shared hosting specifically represents the challenge of dealing with a large number of domains each owned by their own tenant. It becomes quickly apparent that for a shared hosting provider to consider HTTPS as a service for its customers, scale matters a lot. The previous section concluded that *Let's Encrypt* rids hosting providers of certificate purchase cost and the need to perform certificate renewal. What has not been described, but comes courtesy of the automation potential of ACME, is the ability to abstract over the inevitable change and turnover. Even if tenants change subdomains, or if composition of domains per server change frequently, automation handles certification request and renewal. This essentially means that feasibility and cost are potentially no different for a shared hosting provider than for the larger (singular) corporate website. Both require the initial investment in deployment, familiarity and recurring cost for updating/patching but no more. It is our believe that this represents a decreasing opportunity cost for shared hosting providers considering deployment.

Last, with reference to the "standard service offering" incentive described earlier, we note that a number of software solutions for the management of shared web hosting environments (Plesk, cPanel) are considering to include support for ACME. If these were ever be turned into default configuration, use of *Let's Encrypt* may become a non-conscious decision on the part of shared hosting providers.

Judging by recent events surrounding *Let's Encrypt* , it has shaken up the market for certificate authorities. One competitor placed a hostile claim on the *Let's Encrypt* trademark[3], while another launched a half-baked product in an attempt to quickly adopt the robot CA model, which was then found to contain serious vulnerabilities [29] and was subsequently retracted.

We will now proceed to discuss existing coverage of *Let's Encrypt* and its business model in existing literature.

2.5 Related work

Though *Let's Encrypt* is a new entrant, the ecosystem for certificates and their use has a long history of being analyzed. Be it through Internet wide scans [16, 25, 9] and the indexing of their results [8], or using Certificate Transparency [23] (introduced in section 3.2) and their overlap [35]. The company W3Techs covers CA marketshare in the Alexa 10M, including IdenTrust (the CA that has cross-signed the *Let's Encrypt* CA root certificates) in its daily (paid) reports. Jones was the first to publish about *Let's Encrypt* adoption in a series of blogs [20], after leaving employment at *Let's Encrypt* where he created the official stats page [19]. The publication of his most recent blog coincided with *Let's Encrypt*'s adoption of his proposal to change *Let's Encrypt*'s display of statistics, which now include domain measurements of adoption. Helme blogged about early uptake in the Alexa 1M ranking [15]. In a parallel effort, Manousis et al. analysed adoption of *Let's Encrypt* through May 2016, discussing geolocation for certified domains, CA switching within the Alexa 1M ranking, active scans and exploration of malicious use by looking at use for malware domains and typosquatting [26]. Also in parallel, EFF has blogged about different adoption metrics of *Let's Encrypt* and the resulting ranking as biggest CA [12], contrasting statistics from W3Techs [13] and Censys[8].

2.6 Summary

Let's Encrypt potentially rids hosting providers of certificate purchase cost and the need to perform manual certificate renewal. The automation potential of ACME brings the ability to abstract over the inevitable change and turnover. Clearly lowering barriers to adoption of encryption technology accessible to citizens can in theory contribute to their resilience against snooping and tracking on the Web. Yet the question is, do the efforts of *Let's Encrypt* actually reach the hosting providers, and if so, how much and how fast? More generally, is a market approach, contrary to the previously tried modalities [24] of norms (e.g. awareness campaigns) or regulation (e.g. prohibition of snooping) be feasible methods? The contents of this section represent analysis and theorizing by the author and a summary of existing research. What is lacking are actual measurement of *Let's Encrypt* uptake among shared web hosters. This research is an attempt to conduct such measurements, thereby evaluating the suitability of the general approach.

Chapter 3

Design & Method

Having addressed item 1 and item 2 based on literature study, we now turn to empirical research to answer item 3. This Chapter covers its design and applied methods. In order, we cover research design, the data sets used and our chosen methodology.

3.1 Research design

We recall that item 3 stated:

> *"Who has been using Let's Encrypt in the first year since its inception?"*

What follows is a decomposition of the main question to be answered via empirical research, resulting in a list of sub questions. We decompose because the main question is hard to answer directly. We therefore employ a divide and conquer tactic: by answering the sub questions, we can use the most important observations thus obtained to attack the larger one.

The word *"using"* in item 3 gives rise to questions of scale. In the first two sub questions, we attempt to provide insight into usage. To get from absolute numbers (q1) to actual scale, we compare the numbers to the total number of domains (q2).

1. How large is *Let's Encrypt* adoption in distinct domains?
2. What percentage of all domains is getting certificates issued?

We proceed to address the *"Who?"* part of the question. Knowing that the mass of users do not self host, it is important to understand whether growth is realized inside or outside the most popular domains. Continuing, we wonder whether growth is realized due to few large (domain concentrated) or many small (domain sparse) users of *Let's Encrypt*. And while at the organization level, what is their business?

3. Are popular domains under- or over-represented in the use of *Let's Encrypt* certification?
4. Does growth originate from large or small organizations?
5. What sectors are responsible for the largest growth?

With the masses in mind, an important sector would be the segment of shared hosting. Are they using *Let's Encrypt*?

6. Does *Let's Encrypt* manage to penetrate the shared hosting segment?

Finally, while it is simple to experiment with free technology, actual adoption requires more effort and trust. After all, you can always stop with little or no loss of investment. Do users of *Let's Encrypt* remain loyal after having tried the technology?

7. Will users who decide to try *Let's Encrypt* remain loyal?
   Stated differently: How long does an Lets Encrypt certified domain stay certified?

These are the questions that we set out to answer with empirical research. As for any type of empirical research, the type of data available hugely influences what you can and cannot do. We describe what is available and what has been used, considering the trade-offs involved. In the following subsections we make explicit the requirements and scope limitations underlying the selection of data and methods.

3.1.1 Requirements

This subsection documents the design requirements that drove the search for and selection of data sources and the methodology based thereupon.

Time coverage. *Let's Encrypt* issued its first certificate on Sept. 14, 2015, soon followed with a public launch on Nov. 16 of the same year. In order to be able to quantify change, this research should cover the period from Sept. 2015. Moreover, we want to identify trends, not merely perform point measurements. Both this requirement and the fact that the research leading to this thesis started in May 2016 drive a need to obtain historical data.

Domain coverage. The available studies on *Let's Encrypt* , including the majority listed in related research, consider only popular websites, either for lack of interest in smaller sites or due to the ease of data collection on the restricted set. We set ourselves the goal of complete coverage for issuance by *Let's Encrypt*.

3.1.2 Scope decisions

Some design decisions were made for the sake of scope. The available time for this study was limited and as a result some could-be requirements became targets for future work (see also section 5.2).

Issuance versus usage The most dominant design choice we made was to focus on issuance of certificates by *Let's Encrypt* versus their use (deployment) after issuance by site operators. This research, by design, does not quantify the use of certificates on websites, though it certainly establishes a upper bound on such use. We defend this decision as follows. To obtain use statistics, one needs to perform either active scanning on the Internet, or instrument browsers of end-users. Both run into the problem of Domain coverage. There are no known accessible data sets at the time of writing resulting from Internet wide scanning that have good coverage of shared hosting (i.e. multiple domains/certs per IP). Though one can (and researchers do) scan the full IPv4 address range for certificates presented, in shared hosting scenario's the protocols require you to ask for specific domains (SNI, covered previously). This makes large scale collection difficult. Active scanning is also impossible to perform retro-actively, which violates our period coverage requirement. Instrumenting browsers of end-users was not feasible within the time limits of this study.

Market share measurements An interesting question to ask is whether *Let's Encrypt* attracts new users or merely existing users from other CAs. In this research, we have decided to focus on Let's Encrypt for the simple reason that complete data is not freely available on all of the market. This connects to the reasoning given previously for *Issuance versus usage*: there are no known data sets resulting from active scans that have good coverage for shared hosting. As we will see later, the alternative to active scanning that we chose does not (yet) have good coverage for all CAs.

Having covered requirements and scope limitations we now turn to the available data sets.

3.2 Data sets

Certificate Transparency logs on *Let's Encrypt* The certificates issued by *Let's Encrypt* are obtained from Certificate Transparency (CT) logs. CT provides a public append-only log of certificate issuance [23]. For *Let's Encrypt* , this is assumed to be the complete set based on its commitment to full publication of all issued certs. All *known logs* [1] included in Google Chrome are used, though *Let's Encrypt* certificates were all available from the subset run by Google. *Let's Encrypt* issued its first certificate on Sept. 2015 and we evaluate one year of certificates based upon CT data (Sept. 2015-2016), thus hitting our coverage requirements. *Let's Encrypt*'s certificates expire every 90 days and we only consider non-expired certificates. For each certificate, we extract one or more fully qualified domain names (FQDNs) from the `subjectAltName` extension. Each FQDN is then reduced to domain form (see section 3.3) and the resulting set deduplicated. We will refer to domains covered by a non-expired certificate reduced to domain form as *Let's Encrypt domains.*

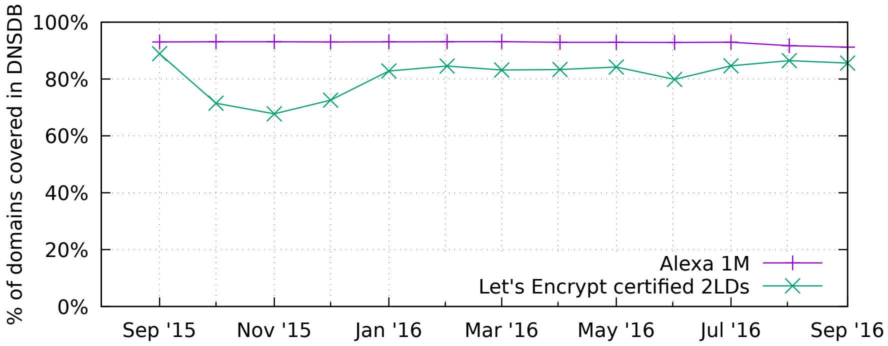


Figure 3: Coverage of DNSDB for Alexa 1M, *Let's Encrypt* certified domains

DNSDB: In order to decompose the *Let's Encrypt* domains in different subsets we use domain information from DNSDB, a passive DNS database that is generously shared with us by Farsight Security. To our knowledge, DNSDB has the best coverage of the overall domain name space that is available to researchers. It draws on hundreds of sensors worldwide and on the authoritative DNS data that various top-level domain (TLD) zone operators publish [10]. We use the subset of A-records in the resulting data set as a monthly drawn sample of all Internet domains, and find the coverage of over 80% of *Let's Encrypt* domains (Figure 3). Though no-one knows exactly how many active domains exist at any point in time, this overlap between disparate sources shows that we use a representative baseline to investigate the coverage of all known domains. In all mappings based on the DNSDB data, records pointing to Martian IP ranges[1] are excluded.

Organizations and organization types: We map the IP addresses obtained from DNSDB into their respective organizations using the methodology described in [32]. This methodology, based on `whois` records, and passive DNS data, also allows us to map IP addresses into various types of providers. We discern between operators of Content Distribution Networks (CDN), Distributed Denial of Service protection (DDoS-protection), end-user Internet Service Providers (ISP), hosting, domain parking and use in education and research networks (EDU).

3.3 Methodology

Absolute and relative growth. *Let's Encrypt* issues certificates with a validity period of 90 days, with contemporary growth numbers in the industry based on certificate count. We measure growth in number of unique domains, reducing the influence of periodic recertification, while increasing the influence of certs with large numbers of embedded FQDNs. To offset the large numbers of subdomains, counts are based on unique domains.

[1] Martians are private and reserved addresses defined by RFC 1918, RFC 5735, and RFC 6598.

With our interest in democratization effects, the assumption here is that distinct domains more likely to indicate distinct people than do distinct subdomains. Moreover, mass deployment for separate subdomains was already possible using wildcard certificates (e.g. `*.example.org`), yet their introduction did not significantly drive overall HTTPS adoption.

An alternative would have been to reduce FQDNs to unique domains not based on known TLD registries, but based on public suffixes. "A public suffix is one under which Internet users can (or historically could) directly register names."[2] This includes the TLD registries (e.g. `example.com`, `example.co.uk`), but notably also `example.wordpress.com`. The disadvantage of deduplication using public suffixes is that this biases counts towards FQDNs with a public suffix and that there is no guarantee that the Public Suffix List is complete, especially for less popular FQDNs. The advantage would be to count the types of shared web hosting that do not give customers unique domains but unique subdomains (such as those provided by Wordpress).

Usage for popular domains. For a measure of popularity, the Alexa 1M ranking was used. The ranking was downloaded on the first day of each month during the period covered and then compared against the set of *Let's Encrypt* domains. Reduced size rankings were derived from the same data, creating top 100K, 10K and 1K rankings. Alexa is frequently used to limit domain coverage in research. On the contrary, we use it merely to enrich the full set of domains in scope and thus measure adoption in the popular segment. Also novel is to measure both absolute contribution to use of *Let's Encrypt* and relative contribution within each ranking.

Big vs. small. To establish the number of certified domains per organization, the set of domains obtained from the certificates is mapped to IP addresses (DNSDB) and then to organizations (IP-org mapping) to produce counts. These are then used to produce empirical cumulative density function plots (ECDFs).

Types of users. By using a mapping between organizations and market segments, the size results of the previous section yield a comparison of uptake among different types of organizations. This methodology is based on previous work by Tajalizadehkhoob et al. [32] and briefly summarized here. We extract domains from DNSDB and the corresponding IP addresses. We then extract from `whois` the netblocks to which these IPs belong and the organizations to which they are assigned. We then merge netblocks belonging to the same organizations. Based on manual mappings and matching the preselected keywords with organization names, we assign organizations to one of the following types: (i) education, (ii) government, (iii) hosting, (iv) Internet service provider (ISP), (v) parking, (vi) DDoS protection, (vii) content delivery networks (CDNs), and (vii) other, e.g. corporate networks such as banks, hospitals, etc. These are used in determining the usage of *Let's Encrypt* by type of organization.

Hosting and shared hosting. The set of DNSDB A-records is used to mark IP space as used for shared hosting. Shared hosting is a boolean property for IP addresses, set to

[2] The Public Suffix List `https://publicsuffix.org/` is a list of all known public suffixes.

true when there are at least 10 distinct domains in the monthly set of A-records pointing to an IP address. This methodology is also based on previous work by Tajalizadehkhoob et al [32]. Having thus marked IPs we proceed to match *Let's Encrypt* domains against this set, again by use of DNSDB.

Certification lifetime. To establish the period of time users of *Let's Encrypt* continue to use the service, we employ survival analysis. Survival analysis is the de-facto statistical method for exactly this purpose (one considers users not renewing certification as not surviving). For each issued *Let's Encrypt* certificate, we obtain the FQDN and the `notBefore` and `notAfter` validity indication fields. FQDNs are used because survival cannot be reliably estimated at the abstraction level of domains. The measurement period is shortened by the final 90 days, because certificates issued during that period have a guaranteed survival. Overlapping validity periods for each FQDN are then joined. To avoid bias in the results for certificates containing many FQDNs, the set of all FQDNs and associated validity periods is then de-duplicated on matching periods (in seconds). The periods are then converted in a lifetime and the Kaplan-Meijer estimator is used to fit a survival function. (Right) censorship events are not shown on the graph for the sake of clarity.

With the methodology for all subquestions described, the next chapter features results and our analysis thereof.

Chapter 4

Results & Analysis

This chapter is structured as described in chapter 3. To gauge *Let's Encrypt*'s contribution to the democratization of encryption technology on Web, we first measure growth in domain coverage (section 4.1). Then, we measure adoption by the most popular websites (and conversely growth outside such rankings, in section 4.2) and the difference between large and small players (section 4.3). We also divide adoption by the type of organization that hosts the certified domains (section 4.4) and examine more specifically the case of shared hosting (section 4.5). Finally, we look at the loyalty of users after first trying *Let's Encrypt* (section 4.6).

Each subsection is structured as follows. We re-iterate the question addressed and formulate a hypothesis. The measurement is then conducted and observcations are made. Finally the question is answered based on these observations.

4.1 Absolute and relative growth

Our first step towards improved understanding of who is using *Let's Encrypt* is to look at growth in terms of domain coverage. How large is *Let's Encrypt* adoption in distinct domains? What percentage of all domains is getting certificates issued? Based on the published number of certificates, we expect growth in both FQDNs and domains. Based on the popularity of issuing certificates for both `{www.,}example.org` the number of certified domains should not trail FQDNs by more than an order of magnitude.

The absolute numbers in Figure 4 show that popularity skyrockets, even on a log scale, and illustrate the growth in FQDNs and domains. From January 2016, the distance between unique number of FQDNs and domains remains relatively constant. Relative coverage (all domains with at least a single FQDN certified) has grown to 2% of all known domains by summer 2016, which puts these numbers in perspective. This is a lower bound on global coverage of *Let's Encrypt* aggregated at second level domains, based on passive DNS data covering $> 80\%$ of thus certified domains. In all, *Let's Encrypt* use is rapidly growing, with the 2% mark both showing the massive scale and the lengthy road ahead.

4.2 Usage for popular domains

Having established that there is growth in number of unique domains we now turn to their popularity. Are popular domains under- or over-represented in the use of *Let's*

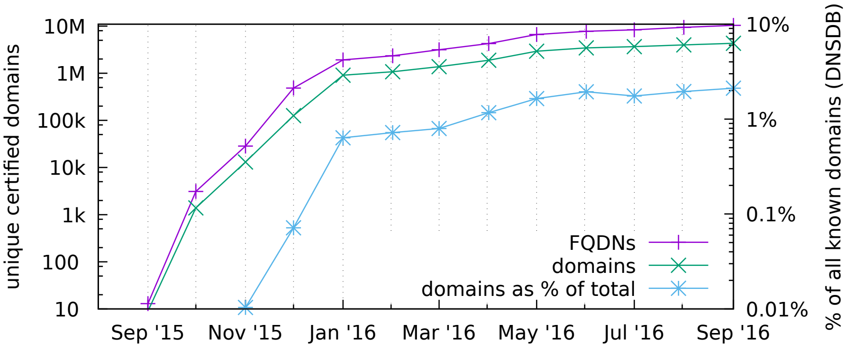


Figure 4: Growth of absolute domain coverage of *Let's Encrypt* for FQDNs, domains. Also plotted is the relative growth of *Let's Encrypt* relative to the total number of domains.

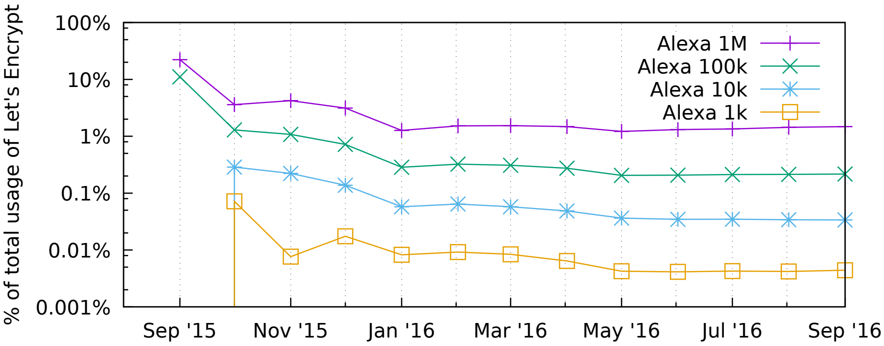


Figure 5: Growth of contribution of Alexa rankings to overall use of *Let's Encrypt* in unique domain counts

Encrypt certification? We distinguish two related questions. First, what is the relative contribution of the top N (for N, an arbitrary ranking)) domains to the total number of *Let's Encrypt* domains? Second, what percentage of domains in the same ranking has had *Let's Encrypt* certs issued? Owners of popular domains likely have more resources, may have an existing relationship with a CA and may want to do more validation than just at the domain level (e.g. extended validation certificates). All in all, enough possible reasons why such domains would be less likely to use *Let's Encrypt*.

Figure 5 shows that the Alexa top 1M domains contribute around 2% of *Let's Encrypt* usage. Contribution to overall usage is necessarily limited by the small subset inherent in the rankings, which the relatively flat profile of the different rankings show. However, Figure 6 shows that usage within the Alexa rankings is steadily growing. Moreover, both issuance and growth thereof are higher than for DNSDB domains, which we previously found to be around 2%. By September 2016, more than 19% of Alexa 1K domains has one or more FQDNs with a *Let's Encrypt* certificate. This holds for larger subsets: near 15% of Alexa 10K, near 9% for Alexa 100K. By the time we get to the Alexa 1M, uptake

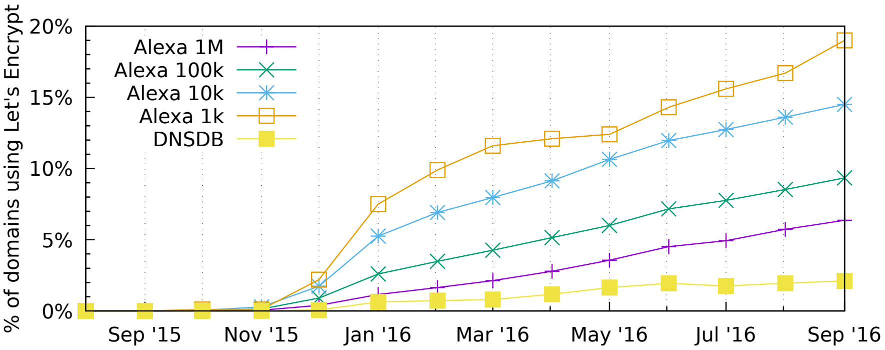


Figure 6: Growth of domain coverage in Alexa ranking. This indicates higher than average usage of *Let's Encrypt* for at least one FQDN under the domain of ranked domains, though not necessarily for the main property.

is at 6%, still 3 times higher than mean use by all DNSDB domains. Now, taking into consideration the fact that a modern web presence usually uses multiple FQDNs and that certificates are also used for non-web services, coverage of a domain in such ranking by no means implies the use of such certificates on their home page. We have verified this to be the case with domains including `wsj.com`, `welt.de` and `lemonde.fr`, which either do not deploy HTTPS or use a different CA at time of writing. Still, our result implies that 19% of the most popular sites know about *Let's Encrypt*'s existence and use its service while having both resources and expertise to deploy and pay for certificates. Are popular domains under or over represented in the use of *Let's Encrypt* certificates? We find that popular domains contribute only a small fraction ($\leq 2\%$) of all *Let's Encrypt* domains, though they show greater relative issuance levels, especially towards the top of the rankings. Growth of *Let's Encrypt* usage is primarily realized outside the popular domains.

4.3 Certificates distribution per organization

The question that is addressed in this section is whether growth originates from organizations responsible for large concentrations of domains or respectively for fewer domains. We define concentration to be large as having a large number of domains pointed to an organization's assigned IP space. Do few large organizations, or large numbers of small organizations account for the majority of uptake? Taking into account the sponsor roster[1], there are quite a few large organizations interested which one would presume want to use the technology they support. Moreover, *Let's Encrypt* solves scalability problems that hurt larger organizations more than it does smaller ones. As a result, we expect the majority of adoption to be contributed by large organizations.

[1] *Let's Encrypt* is run by Internet Security Research Group (ISRG), a non-profit, which takes sponsorship from a number of for-profit entities.

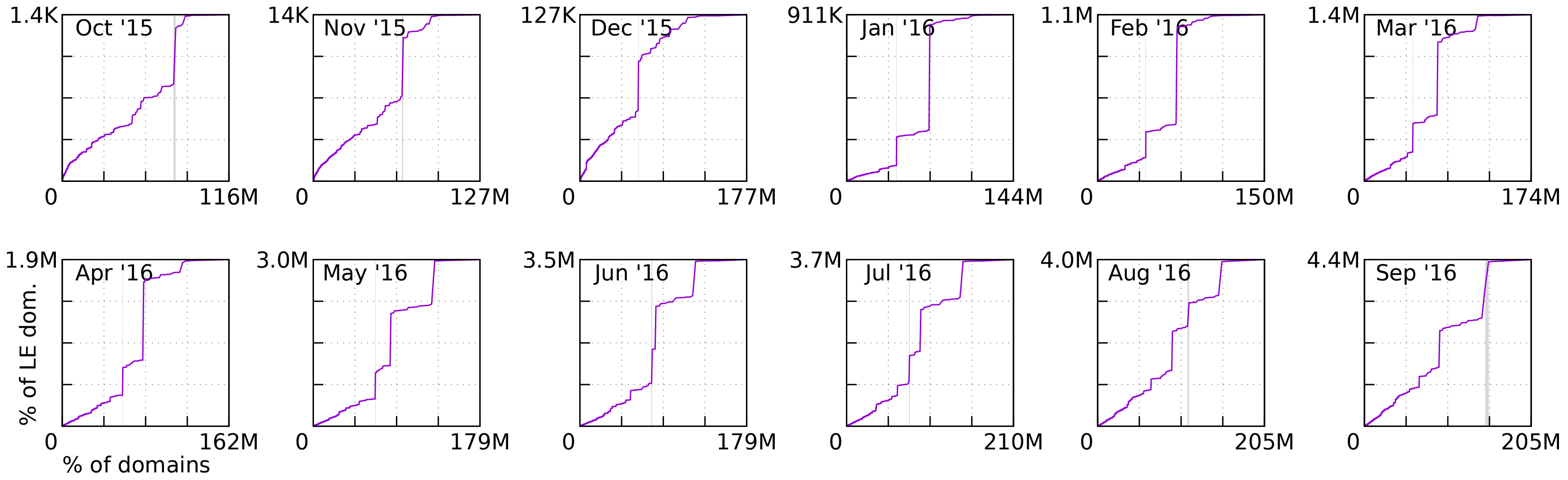
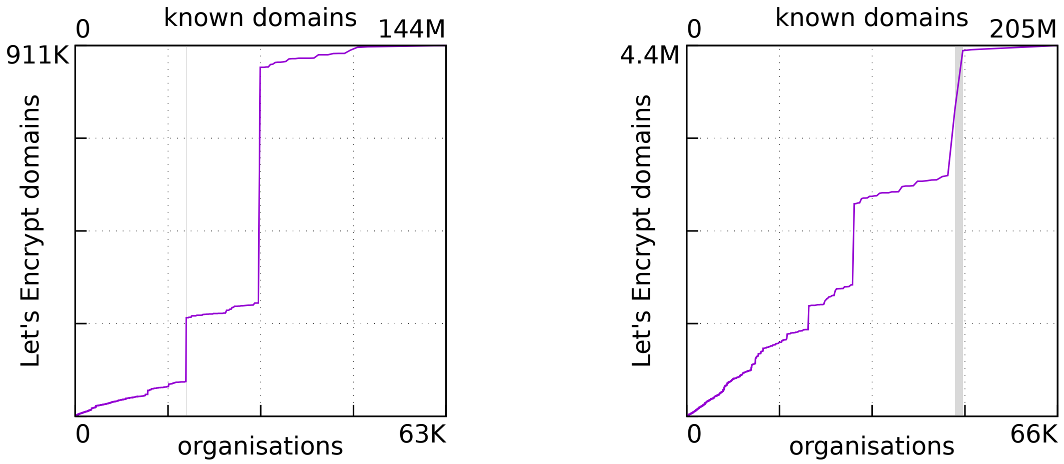


Figure 7: ECDF of *Let's Encrypt* use versus organization size (domain density, measured in number of associated domains). The x-axis (organizations sorted by domain density in ascending order) lists the total number of domains they cover. The y-axis represents the total number of *Let's Encrypt* domains issued that month. A shaded area shows domains not successfully attributed to an organization.



(a) January 2016

(b) September 2016

Figure 8: Two months from Figure 7 in detail. Here, the x-axis has organizations sorted by domain density in ascending order. The added x2-axis represents the total number of known domains (DNSDB).

Figure 7 shows ECDF of *Let's Encrypt* certificates per organization for each month during our measurement period. Figure 8 has more detail for for two selected months of issuance. Steps in these figures indicate bulk issuance of certificates to a particular organization. For example, in January 2016, we see the large vertical line corresponding to deployment at Automattic ($x = 0.5, \Delta y = 63.5\%$), which is especially noticeable compared against November 2016. Automattic is the parent company of wordpress.com, which announced adoption by April 2016[2]. In Figure 7, we can see that increased uptake by more organizations slowly decreases the massive effect of bulk switches over time (by Aug 2016, the profile is noticeable less ragged). By September 2016 (Figure 8b), we can observe three clear steps: Shopify ($x = 0.33, \Delta y = 6\%$), Automattic/wordpress.com ($x = 0.45, \Delta y = 22\%$) and OVH ($x = 0.7, \Delta y = 19\%$). All three have announced issuance for their customers and are jointly responsible for 47% of *Let's Encrypt* certified domains. It is exactly these companies, serving numerous, smaller customers that would otherwise not enable the use of encryption by their visitors.

We find more evidence that suggests a diverse user base. Among 14K organizations that have at least one domain certified with *Let's Encrypt* in Sept 2016, 12K have 50 or fewer domains certified. There is a long tail indeed: 11K of those same organizations have 10 or fewer and 9K have 5 or fewer domains certified. This corresponds to the lower left quadrant of Figure 8b, where smaller organizations jointly responsible for 33% of known domains account for 23% of all *Let's Encrypt* domains. We conclude that *Let's Encrypt* reaches a very broad audience, first based on dominant adoption in

[2] https://en.blog.wordpress.com/2016/04/08/https-everywhere-encryption-for-all-wordpress-com-sites/

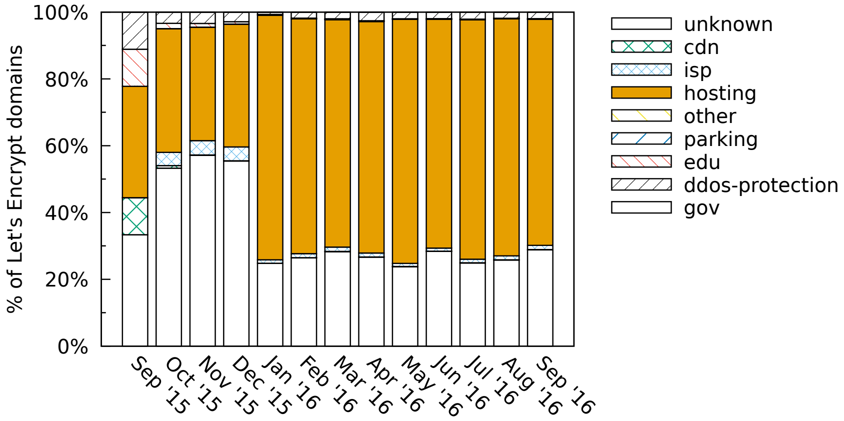


Figure 9: Usage of *Let's Encrypt* by type of organization (in % of domains)

shared hosting (large organizations) and second due to uptake by a large number of organizations with lower domain concentration (small organizations).

4.4 Types of organizations

With multiple subquestions addressed, we now turn to the type of organizations using *Let's Encrypt*. As explained in section 3.2, using a categorization of IP space we quantify the absolute contribution to the total number of certified domains. We expect to see relative low usage by ISPs. Until search rankings in a major way by lack of HTTPS deployment, it also expected that parked domains have little incentive to deploy. Consequently, we expect lower contributions from those categories, especially compared against the total volume. Categories with higher expected contributions are hosting, CDN and DDoS-protection services. First of all because these categories have a high chance of hosting actual services. More specifically because these categories have a specific focus on the web and are therefore a prime target for deployment of HTTPS, requiring certificates.

In Figure 9 we observe the overwhelming majority of domains are associated with hosting, as expected. Contrary to prior expectation, however is that the share of CDN and DDoS protection seems low. In Sept 2016, 68% is hosting, 2% is DDoS protection and less than .1% CDN. This potentially means that there is quite some potential for CDN deployment, seeing how some of the large players sponsor *Let's Encrypt*, yet seem underrepresented in the statistics. These results must be offset against the knowledge that 29% of all domains were not attributed to any of the other categories ('unknown').

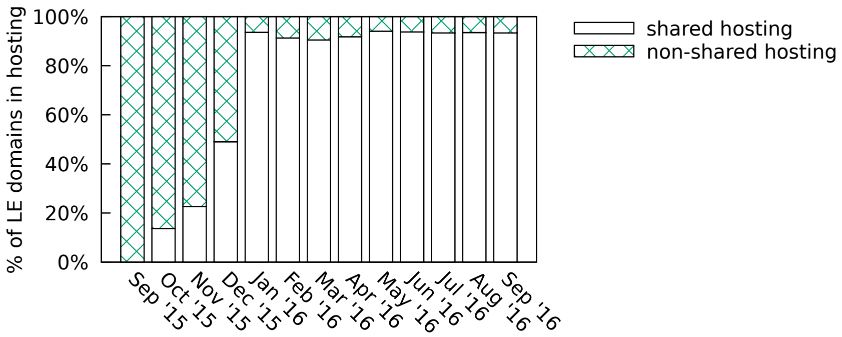


Figure 10: Usage of *Let's Encrypt* within hosting segment: shared vs. non-shared (in % of domains)

4.5 Hosting and shared hosting

With hosting identified as the largest category of *Let's Encrypt* use, we now focus on the specific segment of shared hosting. Does *Let's Encrypt* manage to penetrate the shared hosting segment? Shared hosting where prices are at their lowest and profit margins are traditionally thinnest would make investment in encryption technology less likely. Moreover, providing free access to *Let's Encrypt* might compete with (re)selling paid certificates. Still, we expect uptake in this market segment due to the disappearing cost factor and the possibility and ease of automation.

Figure 10 is a histogram of relative market share within the hosting segment, split between shared and non-shared hosting models. We find that from Jan 2016, *Let's Encrypt* use within hosting is dominantly connected to shared hosting models, with a penetration above 90%. Recalling that by Sept. 2016 the overall hosting segment is dominant (67%), we find that *Let's Encrypt* has very high overall utilization in shared hosting, which has traditionally been the least likely candidate for adoption of encryption.

4.6 Certification lifetime

Another interesting question we address is whether users who decide to try *Let's Encrypt* remain loyal. In other words, how long does an *Let's Encrypt* certified domain stay certified? After all, with issuance pricing at zero, it could be the case that *Let's Encrypt* certificates are only used for one-time try-out of the technology. We identify three components that are likely to influence the outcome to this question: *(i)* automation working correctly; with validity limited to 90 days, not having automation set-up likely causes renewal failure, *(ii)* user satisfaction with the service and its certificates, *(iii)* if the domains being certified by *Let's Encrypt* actually meant to be long-lived.

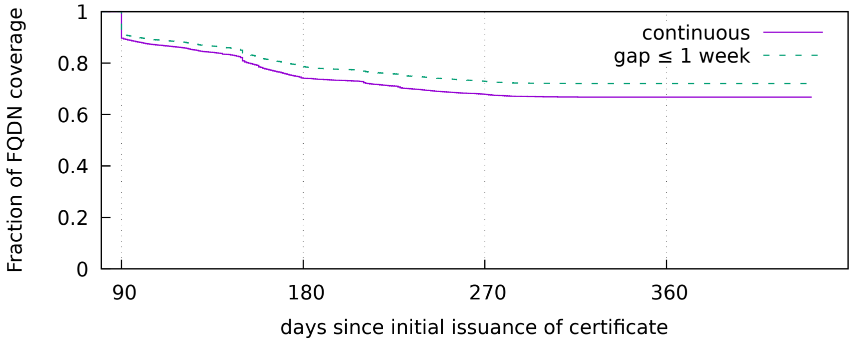


Figure 11: Survival analysis of certificate renewal

Figure 11 shows the estimated survival function of *Let's Encrypt* certified FQDNs featuring two functions. The continuous function measures survival without any downtime: survival implies the issuance of certificates with perfectly overlapping validity periods. The second function measures survival with a maximum 1 week gap in between consecutive validity periods. This accounts among other things for failure in automation, corrected after the previous certificate expires. We observe 100% coverage until 90 days due to default validity period of that length. After those 90 days we see the expected drop: domains that either stop being certified, where automation was not successful or that expired. The survival curve noticeably flattens after $x = 270$, which shows the effectiveness of automation. The likeness between $gap = 0$ and $gap \leq 1$ week shows that beyond initial downtime, further survival is roughly similar. This may be explained by users that get continuous coverage after successful setup of automation. With $\geq 70\%$ FQDN coverage after a full year, we can conclude that the overwhelming majority of *Let's Encrypt* users remain loyal to the service during our measurement period. Keeping in mind the size (section 4.3) and type (section 4.4) of users (dominantly big hosting providers), this is not surprising.

Chapter 5

Discussion

This chapter discusses the outcomes of the empirical study performed in chapter 4. It answers item 4: *"What insights do the results from the case study on Let's Encrypt provide for bringing HTTPS to the masses?"* This chapter is structured in two parts. First, we discuss insights from the results and future potential for the growth of HTTPS deployment. Then, we discuss insights for future research and cover the areas that we deem of interest for future work.

5.1 Insights resulting from the Let's Encrypt case study

With 2% growth in coverage of known domains (DNSDB), section 4.1 has shown a clear market for both *Let's Encrypt* and the ACME protocol. From chapter 2 we recall that the main contributions of *Let's Encrypt* were cost-free issuance and automation of the request and renewal process. Either one or both of these factors have proven to be capable of overcoming deployment lag for a still small (4.4M) but fast growing chunk of domain space. And there is reason to assume that the trend of growth will continue.

In the next subsection, we will discuss potential for HTTPS deployment growth in the wake of *Let's Encrypt*. We will then discuss further potential effects not directly relating to growth. Finally, we zoom out one level and regard the potential for further collective action in information security modeled after the sponsor model of *Let's Encrypt*.

5.1.1 The potential for HTTPS deployment growth

In this subsection, we reflect upon the potential for further growth in HTTPS deployment. We cover both growth for *Let's Encrypt* itself and growth in general.

Sectors where automation has potential In section 4.4 the different market segments were contrasted with respect to uptake of *Let's Encrypt*. We noted that a number of segments were intuitively behind in their deployment based on their automation potential not currently realizable with other CAs. Notably, this includes the market for DDoS protection and CDN services. Both cover a changing set of customer domains, much like shared hosting, with the resulting churn a prime candidate to be addressed by automation. A final candidate, though perhaps with less societal utility, is adoption in the domain parking sector. Whenever SEO incentives increase, the potential for automation would make it relatively cheap to start serving the ads on parked domains over HTTPS.

Sectors where pricing has potential Another set of segments that saw reduced uptake in section 4.4 were domains associated with government and education. These segments are much smaller in terms of domain space and might have less potential for automation. In contrast, here the zero cost argument could result in future adoption, empowering individuals to deploy HTTPS where following a procurement process was necessary in the past. The author feels that the price argument is less convincing than the previous corresponding paragraph on automation.

ACME clients: improving survival It is one thing to note areas for further growth, yet retaining active users will become increasingly important. In section 4.6 we identified a drop in uptake around the first and second renewal periods. The author believes that continued development on the ACME clients for certificate request and renewal has the potential to (further) reduce this drop. And though churn, failures and users transitioning to other CAs will always continue, anecdotal evidence suggests that usability, diversity and stability of available software has increased to that effect.

ACME clients: integration in software In section 2.4, we briefly discussed the potential for integration of ACME clients in existing (popular) software. *Let's Encrypt* has shown growth and stability in its first year. And integration in software takes time, but also trust that the additional complexity and maintenance burden is worth carrying. The amount available software with built-in or third party add-ons to request and deploy certificates out of the box is expected to increase as a result. And when these are turned on in default configurations, we may expect growth in HTTPS adoption.

Word of mouth In section 4.2 we discussed that 19% of the world's most visited websites (Alexa) had at least one certificate issued. This is evidence of the fact that *Let's Encrypt* is getting noticed in the upper echelons. Over time other segments of the Internet (and world) are bound to learn about *Let's Encrypt* , further driving deployment.

Competing CAs Finally, other CAs –irrespective of any growth realized by *Let's Encrypt*–, are bound to take note of its rapid growth. First because *Let's Encrypt* is showing that there is a market currently not served by the CAs, namely that of smaller organizations and shared hosting providers. But also because the automation potential in ACME, available as a public standard [7], could also be of interest for CAs continuing to charge for their services.

5.1.2 Other potential effects

Having discussed the growth potential for HTTPS deployment in light of the first year of *Let's Encrypt*, we will now consider other potential effects.

Competition in the CA market Related work on a smaller set of domains has shown *Let's Encrypt*'s growth to be dominantly in new entrants to the certificate market and only to a limited extent at the cost of competitors [13]. Still, other CAs are likely to follow *Let's Encrypt* course with close attention. This may bring innovation and differentiation to a market which has largely had none [5], potentially to the collective benefit of all.

CA/Browser forum The CA/Browser forum[1] is a group of CAs, browser and related software vendors that "advances industry best practices to improve the ways that certificates are used to the benefit of Internet users and the security of their communications". Part of these best practices have been adopted as formal criteria for the inclusion of a CA's trust anchor in browsers, essentially making them mandatory rules for the CA industry. Most members of the CA/B Forum are for-profit entities, with few exceptions. The rise of *Let's Encrypt* brings another non-profit to this governance body, which given its stated aims may well have positive effect on future industry standards.

Cryptographic agility The automation potential of ACME and its growing uptake of clients may in the future help deployment of new cryptographic primitives. While the transition between SHA1 and SHA256 is underway, a similar move may one day be necessary for RSA and ECDSA. By controlling the auto-update mechanism for clients, its authors (not necessarily *Let's Encrypt*) may help expedite such transitions, with a potential for improved cryptographic agility as a result.

5.1.3 The potential for collective action in security

As covered in chapter 2, the business model of *Let's Encrypt* is different in the sense that it does not directly depend on the volume of certificate issuance for its income. Multiple sponsors contribute annual funding, amounting to a yearly operating cost of 2.9M$ budgeted for 2017 [4]. Though its sponsors are definitely served by its work, collectively they are also addressing a market that a for-profit CA would not consider to be in its best interest. This idea is further illustrated by the fundraiser started by *Let's Encrypt* to get the public to contribute a share of its operating cost[2].

[1] `https://cabforum.org/`

[2] `https://letsencrypt.org/2016/11/01/launching-our-crowdfunding-campaign.html`

ISRG/*Let's Encrypt* is not the first non-profit body thus contributing to Internet security. Recent examples include such efforts as the Linux Foundation's Core Infrastructure Initiative[3], that "enables technology companies to collaboratively identify and fund open source projects that are in need of assistance, while allowing the developers to continue their work under the community norms that have made open source so successful." These efforts have the potential to break the effects of externalities, i.e. everyone only caring about their immediate own interest, by collectively producing the public good of increased resilience on the Web.

Let's Encrypt has proven this model to be viable for the CA market, which is promising for other areas of security where similar opportunities may exist. Perhaps similar business models can be used to solve problems in such areas as the software update problem for IoT devices.

5.2 Future work

We conclude this chapter with insights for future work and areas to extend this research in future efforts.

5.2.1 Insights for future work

Certificate Transparency We have found CT to house a wealth of information useful for research. In light of Google's market pressure on CAs to adopt logging practices [30], this source of information will only grow in the future. For research on X.509 certificates, the HTTPS deployment effort or even more general SSL/TLS research, CT holds great promise for the future as a publicly accessible, auditable source of information.

Lack of SNI-aware certificate data sets Future research analyzing the growth of certificate coverage for the shared hosting sector will run into the same lack of data that we faced in this work. This is the lack of historical certificate data sets that not only cover the IPv4 address range, but also includes multiple certificates per IP (use of SNI). The lack of such data sets make it hard to perform comprehensive market studies likely of interest to identify further potential barriers to HTTPS adoption or successes in removing them.

Authoritative 2LD lists This research has built upon a (reduced) public suffix list for 2LD measurement. It would be great if this data set would become a publicly maintained resource, for example by tagging the available data in categories (ISP provided, registrar provided). This would preclude the need to manually maintain this list, which is necessarily error prone.

[3] `https://www.coreinfrastructure.org/`, quote taken from FAQ page.

5.2.2 Potential extensions to this research

The HTTPS ecosystem and deployment challenges in general seem to be a fertile ground for further (empirical) study. During the course of our research, we have identified a number of topics that could serve as potential extensions, or that were wholly different but not less interesting. We concisely list a number of topics for further study.

Use versus issuance statistics One particular scoping decision affecting this work was the choice to measure issuance, not actual deployment. It would be very interesting to start measuring both and to compare trends. Perhaps what is learned in the future may reflect on developments we have uncovered about the past.

Survival analysis correlated to ACME client release It would appear that there is more to be gained from studying the issuance patterns of *Let's Encrypt*. One such extension would be to correlated the survival analysis against releases of popular ACME clients in order to identify the impact of improvements made and perhaps thus identify further enhancements possible.

Public suffix vs. 2LD With a number of big web hosters known to offer subdomains to their users, it would be very interesting to re-run the empirical study performed here with domains aggregated at the public suffix level minus one in addition to the aggregation on 2LD we have chosen to perform. This would increase visibility on such practices, covering a currently unknown size of market at an even lower price point.

The internationalization angle Though this research has not focused on the geographic dispersion of *Let's Encrypt* users, the very recent introduction of Internationalized Domain Name (IDN) support[4] gives rise to the question: what is the influence on uptake in countries non-native to the ASCII data set. After all, one can only talk about true democratization when considering a truly worldwide scale, which ASCII was not designed to convey.

Abuse What has also not been covered in this research is the *mis*use angle for *Let's Encrypt*. How much is *Let's Encrypt* employed for such malpractices as phishing scams, malware distribution or similar abuse? Research in that direction should balance evidence of misuse against the effectiveness of measures taken by *Let's Encrypt* (e.g. revocation speed), providing facts for the popular discussion on the use of encryption for less lofty goals in society.

[4] https://letsencrypt.org/2016/10/21/introducing-idn-support.html

Chapter 6

Conclusions

In this final section, we bring together the answers to the various questions posed in chapter 1, collectively addressing the main research question. We treat each subquestion in order.

1. *"What prevents widespread adoption of HTTPS?"*

   The overwhelming majority of end-users, the masses, do not self-host content on the Web, instead making use of hosting providers. These hosting providers have equal or greater impact on whether or not an end-user is able to communicate with preservation of confidentiality.

   We find both positive and negative incentives that affect deployment by shared hosting providers (chapter 2). On the negative side there are cost and complexity barriers that affect deployment. Deployment statistics suggest that the negative incentives outweigh their positive counterparts, delaying widespread deployment. We show that these effects are especially relevant for the segment of shared hosting.

2. *"How does Let's Encrypt contribute to widespread adoption?"*

   Let's Encrypt's approach contains two novel contributions (chapter 2). First, *Let's Encrypt* removes the purchase cost for hosting providers by not charging for certificate issuance or revocation. Second, use of *Let's Encrypt* alleviates the need for manual certificate request or renewal by use of a standardized protocol (ACME). ACME allows for use of client software with a one-time setup cost –relevant for individuals and small organizations– or that scales to large numbers of domains –relevant for (shared) hosting providers–. Both are differentiators against large parts of the current market for Certificate Authorities.

3. *"Who has been using Let's Encrypt in the first year since its inception?"*

   We have performed empirical research into the first year of issuance for *Let's Encrypt*, with the methodology described in chapter 3. The main findings from our analysis in chapter 4 are as follows:

   a) Use of *Let's Encrypt* has clearly taken off. We find coverage of 2% of all known domains (DNSDB), a lower bound on global coverage of *Let's Encrypt* domains.

   b) The bulk of issuance is for less popular domains (outside the Alexa 1M ranking of most popular websites). However, 19% of the most popular sites (Alexa 1K) have had at least one certificate issued in their domain. In other words, *Let's Encrypt* is also being employed by sites that have both resources and expertise to deploy and pay for certificates.

c) Dominant drivers of *Let's Encrypt*'s growth are hosting companies (68%), in particular those bulk certifying domains of their users (3 companies cover 47%). It is exactly these companies, serving numerous, smaller customers that would otherwise not enable the use of encryption by their visitors.

d) For the segment of hosting, over 90% of domains certified are connected to shared hosting, which has traditionally been the least likely candidate for adoption of encryption.

e) The majority ($\geq$ 70%) of *Let's Encrypt* users remain loyal to the service during the measurement period.

4. *"What insights do the results from the case study on Let's Encrypt provide for bringing HTTPS to the masses?"*

   With 2% growth in coverage of known domains (DNSDB) and a dominant representation of both large shared hosting providers and numerous small organizations, we have illustrated a clear market for both *Let's Encrypt* and the ACME protocol. Either one or both of these factors have proven to be capable of overcoming deployment lag for a still small (4.4M) but fast growing chunk of domain space. Though there is much potential for further improvement, our results indicate that the operational model of *Let's Encrypt* holds promise for a) further growth of HTTPS adoption b) more generally the potential for collective action in security and c) a number of related positive effects.

With the subquestions covered, we now turn to the main question:

> *"How to bring HTTPS to the masses?"*

Per our coverage of future research (section 5.2) many challenges remain and *Let's Encrypt* is far from ubiquitous. Yet with the results presented, it is evident that the approach pursued by *Let's Encrypt* in its first year of operation is worthy of broader consideration and adoption in the industry. Though issuing for free will not fit the majority of business models, the adoption of ACME for automation and future integration with server software may prove pivotal to bring HTTPS to the masses.

Stated more generally, we find that collective action through sponsored non-profit organizations such as ISRG/*Let's Encrypt* may hold the key to related problems in security that are not currently being solved. Taking a market approach to the problem of deploying technical mitigations may be well worth future consideration.

Acknowledgements

The author would like to acknowledge the contribution and support from a number of people, some of them mentioned here.

First of all, Annelieke, for your patience during the many hours of research and writing. Mireille and Jan, thank you for allowing me to freely pursue my research interest and find those to guide me closest to my chosen topic. Maciej and Giovane, I am very grateful for your enthousiasm, initially when I popped (back) up out of nowhere, during a period of sparse status updates with limited results and finally when collaborating on what we now hope will become a joint publication. I thoroughly enjoyed our interaction, be it about the research, the California weather or the curious waking hours of a new dad. Jan, thank you for your insights and feedback helping me to broaden the scope of my research, especially since your time was understandably limited. Finally, at Deloitte, Marko, Annika and JJ for making it possible to pursue a second master's degree.

Copyright

Bibliography

[1] Certificate Transparency - Known logs. https://www.certificate-transparency.org/known-logs.

[2] Google transparency report - https - https usage. https://www.google.com/transparencyreport/https/metrics/?hl=en.

[3] Josh Aas. Defending our brand. https://letsencrypt.org/2016/06/23/defending-our-brand.html.

[4] Josh Aas. What it costs to run let's encrypt. https://letsencrypt.org/2016/09/20/what-it-costs-to-run-lets-encrypt.html.

[5] Hadi Asghari, Michel Van Eeten, Axel Arnbak, and Nico Van Eijk. Security economics in the https value chain. In *Twelfth Workshop on the Economics of Information Security (WEIS 2013), Washington, DC*, 2013.

[6] Zineb Ait Bahajji and Gary Illyes. HTTPS as a ranking signal. https://webmasters.googleblog.com/2014/08/https-as-ranking-signal.html, August 2014.

[7] R. Barnes, J. Hoffman-Andrews, and J. Kasten. Automatic Certificate Management Environment (ACME). draft-ietf-acme-acme-03, July 2016.

[8] Zakir Durumeric, David Adrian, Ariana Mirian, Michael Bailey, and J. Alex Halderman. A search engine backed by Internet-wide scanning. In *Proc. of ACM CCS*, 2015.

[9] Zakir Durumeric, James Kasten, Michael Bailey, and J Alex Halderman. Analysis of the https certificate ecosystem. In *Proc. of IMC*, pages 291–304, 2013.

[10] Fairsight. DNSDB. https://www.dnsdb.info/.

[11] S. Farrell and H. Tschofenig. Pervasive Monitoring Is an Attack. RFC 7258 (Best Current Practice), 2014.

[12] Gennie Gebhart and Seth Schoen. Is lets encrypt the largest certificate authority on the web? https://www.eff.org/deeplinks/2016/10/lets-encrypt-largest-certificate-authority-web, Oct 2016.

[13] Matthias Gelbmann. The impact of let's encrypt on the ssl certificate market. https://w3techs.com/blog/entry/the_impact_of_lets_encrypt_on_the_ssl_certificate_market, Sep 2016.

[14] Ilya Grigorik. TLS has exactly one performance problem: it is not used widely enough. Everything else can be optimized. https://istlsfastyet.com/, 2016.

[15] Scott Helme. Security headers in the alexa top 1 million - let's encrypt usage. https://scotthelme.co.uk/security-headers-alexa-top-million/, Feb 2016.

[16] Ralph Holz, Lothar Braun, Nils Kammenhuber, and Georg Carle. The ssl landscape: a thorough analysis of the x. 509 pki using active and passive measurements. In *Proc. of IMC*, pages 427–444, Nov 2011.

[17] ISRG. Internet Security Research Group (ISRG). https://letsencrypt.org/isrg/, May 2016.

[18] ISRG. Let's encrypt - About. https://letsencrypt.org/about/, May 2016.

[19] ISRG. Let's encrypt stats. https://letsencrypt.org/stats/, 2016.

[20] J.C. Jones. Blog series on growth of Let's Encrypt. https://tacticalsecret.com/tag/letsencrypt/, 2016.

[21] James Douglas Kasten Jr. *Server Authentication on the Past, Present, and Future Internet*. PhD thesis, The University of Michigan, 2015.

[22] Paul Kinlan. Geolocation api removed from unsecured origins in chrome 50. https://developers.google.com/web/updates/2016/04/geolocation-on-secure-contexts-only.

[23] B. Laurie, A. Langley, E. Kasper, E. Messeri, and R. Stradling. Certificate Transparency. RFC 6962-bis-19 (Internet-Draft), August 2016.

[24] Lawrence Lessig. *Code*. Lawrence Lessig, 2006.

[25] Olivier Levillain, Arnaud Ébalard, Benjamin Morin, and Hervé Debar. One year of ssl internet measurement. In *Proc. of ACSAC*, pages 11–20. ACM, Dec 2012.

[26] Antonis Manousis, Roy Ragsdale, Ben Draffin, Adwiteeya Agrawal, and Vyas Sekar. Shedding light on the adoption of let's encrypt. *arXiv preprint arXiv:1611.00469*, 2016.

[27] P.V. Mockapetris. Domain names - concepts and facilities. RFC 1034 (Internet Standard), November 1987. Updated by RFCs 1101, 1183, 1348, 1876, 1982, 2065, 2181, 2308, 2535, 4033, 4034, 4035, 4343, 4035, 4592, 5936.

[28] Huib Modderkolk. Kabinet houdt vast aan massaal aftappen internetverkeer (Dutch). http://www.volkskrant.nl/media/kabinet-houdt-vast-aan-massaal-aftappen-internetverkeer~a4291392/, 2016.

[29] Christiaan Ottow. Startencrypt considered harmful today. https://www.computest.nl/blog/startencrypt-considered-harmful-today/.

[30] Ryan Sleevi. Sustaining digital certificate security. https://security.googleblog.com/2015/10/sustaining-digital-certificate-security.html.

[31] WAM Steenbruggen et al. Publieke dimensies van privé-communicatie: een onderzoek naar de verantwoordelijkheid van de overheid bij de bescherming van vertrouwelijke communicatie in het digitale tijdperk. 2009.

[32] Samaneh Tajalizadehkhoob, Maciej Korczyński, Arman Noroozian, Carlos Gañán, and Michel van Eeten. Apples, oranges and hosting providers: Heterogeneity and security in the hosting market. In *Proc. of NOMS*, Apr 2016.

[33] Kurt Thomas, Elie Bursztein, Chris Grier, Grant Ho, Nav Jagpal, Alexandros Kapravelos, Damon McCoy, Antonio Nappa, Vern Paxson, Paul Pearce, et al. Ad injection at scale: Assessing deceptive advertisement modifications. In *Proc. of IEEE S&P*, pages 151–167, 2015.

[34] Narseo Vallina-Rodriguez, Srikanth Sundaresan, Christian Kreibich, and Vern Paxson. Header enrichment or isp enrichment?: Emerging privacy threats in mobile networks. In *Proceedings of the 2015 ACM SIGCOMM Workshop on Hot Topics in Middleboxes and Network Function Virtualization*, pages 25–30. ACM, 2015.

[35] Benjamin VanderSloot, Johanna Amann, Matthew Bernhard, Zakir Durumeric, Michael Bailey, and J Alex Halderman. Towards a complete view of the certificate ecosystem. In *Proc. of IMC*, pages 543–549, Nov 2016.

[36] Samuel D Warren and Louis D Brandeis. The right to privacy. *Harvard law review*, pages 193–220, 1890.

[37] Alan F Westin. Privacy and freedom. *Washington and Lee Law Review*, 25(1):166, 1968.