Linköping University | Department of Computer and Information Science
Bachelor's thesis, 16 ECTS | Informationsteknologi
2020 | LIU-IDA/LITH-EX-G--20/061--SE

Relay Racing with X.509 Mayflies

– An Analysis of Certificate Replacements and Validity Periods in HTTPS Certificate Logs

Stafettlöpning med X.509-dagsländor: En Analys av Certifikatutbyten och Giltighetsperioder i HTTPS-certifikatloggar

Carl Magnus Bruhner
Oscar Linnarsson

Supervisor : Niklas Carlsson
Examiner : Marcus Bendtsen

Linköpings universitet
SE–581 83 Linköping
+46 13 28 10 00 , www.liu.se

Upphovsrätt

Copyright

Students in the 5 year Information Technology program complete a semester-long software development project during their sixth semester (third year). The project is completed in mid-sized groups, and the students implement a mobile application intended to be used in a multi-actor setting, currently a search and rescue scenario. In parallel they study several topics relevant to the technical and ethical considerations in the project. The project culminates by demonstrating a working product and a written report documenting the results of the practical development process including requirements elicitation. During the final stage of the semester, students create small groups and specialise in one topic, resulting in a bachelor thesis. The current report represents the results obtained during this specialisation work. Hence, the thesis should be viewed as part of a larger body of work required to pass the semester, including the conditions and requirements for a bachelor thesis.

Abstract

Certificates are the foundation of secure communication over the internet as of today. While certificates can be issued with long validity periods, there is always a risk of having them compromised during their lifetime. A good practice is therefore to use shorter validity periods. However, this limits the certificate lifetime and gives less flexibility in the timing of certificate replacements.

In this thesis, we use publicly available network logs from Rapid7's Project Sonar to provide an overview of the current state of certificate usage behavior. Specifically, we look at the Let's Encrypt mass revocation event in March 2020, where millions of certificates were revoked with just five days notice. In general, we show how this kind of datasets can be used, and as a deeper exploration we analyze certificate validity, lifetime and use of certificates with overlapping validity periods, as well as discuss how our findings relate to industry standard and current security trends. Specifically, we isolate automated certificate services such as Let's Encrypt and cPanel to see how their certificates differ in characteristics from other certificates in general.

Based on our findings, we propose a set of rules to help improve the trust in certificate usage and strengthen security online, introducing an *Always secure* policy aligning certificate validity with revocation time limits in order to replace revocation requirements and overcoming the fact that mobile devices today ignore this very important security feature. To round things off, we provide some ideas for further research based on our findings and what we see possible with datasets such as the one researched in this thesis.

Acknowledgments

The authors would like to thank the Rapid7 research team, both for collecting and providing the SSL certificate data as well as for helping with additional questions and raised data limits. We similarly thank Let's Encrypt for providing such vivid transparency regarding the mass-revocation event, and for answering our questions and engaging in providing additional data for our research.

We would like to thank Martin Arlitt, Adjunct Assistant Professor at the University of Calgary, for his input regarding the Rapid7 dataset and for reviewing our thesis and helping us analyze our results.

Also, thanks to our fellow students Carl Ekblad, Alexandra Goltsis, Henrik Wendt and Matteus Henriksson for reviewing our thesis and providing valuable feedback to improve it.

Last but not least, we thank our tireless supervisor Niklas Carlsson, for coping with our endless questions, providing academic depth and creativity as well as spending several late nights online at Zoom guiding us forward during the pandemic reality of 2020.

Contents

List of Figures

List of Tables

1 Introduction

Do you trust your encrypted connections to be secure, and all compromised certificates used for encryption to be securely revoked? That just might be a bad idea. Today, a large majority of the most popular websites uses encrypted data traffic between server and user [2]. Only a few years ago, this was not the case, but thanks to initiatives such as Chromium gradually marking non-encrypted websites as *"not secure"* [37, 48] and Mozilla's intents to deprecate what they call *"non-secure HTTP"* [7] there has been a steady growth of HTTPS (Hypertext Transfer Protocol Secure) adoption [22] as well as adoption of other encryption protocols such as HTTP/2, QUIC and Facebook Zero [53]. HTTPS encryption uses key data from X.509 certificates (see more in Section 2.4) that are either issued by a certificate authority or self-signed. Historically, certificates have been issued with long validity periods, but as shorter periods for more or less obvious reasons are more secure [52] industry standards and recommendations [9] and initiatives such as the 90-days-alive certificates of Let's Encrypt [2] have contributed to the continued shortening of average lifespan. However, there does not seem to be any general replacement standards or policies [24].

This paper aims at analyzing the current state of HTTPS certificate usage from network scan logs in order to, with guidance from industry standards, provide recommendations on best usage and suggestions about feasible certificate lifetime and replacement policies.

In Section 2.10, there are some useful definitions that are established in this thesis in order to be aid with interpretation and discussion.

1.1 Motivation

Surprisingly, based on extensive searching, the area of certificate overlapping, replacement strategies, validity and longevity does not seem to have been researched much. The main idea of this paper is therefore to explore this area and seek to find patterns in certificate replacement data compiled from the publicly available Project Sonar SSL dataset. The findings will be connected with certificate data such as, but not limited to, issuer, subject, types, validity periods, lifetime, and more in search of possible explanatory models to be investigated.

1.2 Aim

The main purpose with this thesis is to see what general trends could be found in a publicly available SSL network scanning dataset open for research and analysis. Additionally, the aim is to see whether there are any connections between certain certificate characteristics and the usage of those certificates during its validity, lifetime and beyond.

1.3 Research questions

At a high level, this thesis aims to extract and compile the most important observations and insights that can be made regarding HTTPS certificate usage from the Project Sonar SSL dataset. As a mean to achieve this objective, we use the processed dataset to answer the following example questions:

1. How often are certificates replaced, and when are they replaced in relation to their validity?

2. How do these replacement patterns vary between different certificate characteristics and properties such as issuer, validity period, and more?

3. What impact did Let's Encrypt's mass revocation event in the beginning of March 2020 have on certificate renewal patterns with regards to overlapping?

Based on the answers to these questions, we then provide insights into how to better manage certificate lifetimes, validity periods, and certificate overlaps to strengthen internet security.

1.4 Contributions

We have extracted and analyzed certificates and their usage from the Project Sonar SSL dataset, and proved that it can be used for analyzing long-term trends as well as shorter event-specific trends. Our analysis focuses on certificates issued with a validity just above the industry-complying requirement of 3.25 years, and the data is aggregated through a multi-step methodology where we (i) parse certificate data, (ii) extract the corresponding first and last observation of each certificate, (iii) sort the data for further analysis, and (iv) calculate the replacement relations to be analyzed.

Our findings show that there is a clear discrepancy in overlapping patterns between the top-issuing Certificate Authorities (CAs), dividing those having automated renewal/replacement support and those dependent of manual work. We also reveal a general overlapping pattern depending on validity periods. Additionally, we look at a recent revocation event concerning the largest-by-number issuing CA Let's Encrypt, revealing a notable increase in overlapping following the revocation.

Finally, based on our findings we provide insights into the current HTTPS certificate management and provide recommendations for improvements, potentially to removing the need of costly revocations. We summarize these recommendations and suggestions as the following:

- *Certificate Validity recommendations*, where a bold validity period of maximum 5 days is presented, based on current state-of-the-art examples put in relation to requirements by industry standards in the event of a revocation.

- *Certificate Replacement suggestions*, focused on following the technical possibilities demonstrated by industry leaders, proving the concept of fully automated certificate replacements to support shorter validity periods.

1.5 Delimitations

This study is based on port 443 data from the SSL subset of Project Sonar's HTTPS study, as this port is the primary port of secure HTTP traffic and thus by far the largest subset of the study. However, as this study is conducted together with other non-standard ports for HTTPS, where full certificate data is only included in the subset of the port where it was first observed [50], there is a possibility that certificates appearing in port 443 logs are not at any time present in the provided certificate data of the port 443 study. Nonetheless, limiting this thesis study to ”standard HTTPS usage”, this should be a reasonable delimitation.

We have also chosen to limit the study to the inclusive period of 2017-04-25 and 2020-04-22, to give three years of available data. Nonetheless, this limits the available data of certificates used in the beginning of the period, whereas the more recent data is limited not by the availability of certificate data (other than the aforementioned port limitation) but by the limited time frame of available usage. This however helps in limiting the processing time needed in order to continue with the analysis.

The Project Sonar study is conducted only on IPv4, thus limiting the resulting data to a subset of the total number of available HTTPS hosts online. However, expanding the scope to include IPv6 is simply not possible within the limits of the study, as will be explained as a short comment in Section 2.1.

1.6 Thesis outline

In order to analyze the use of X.509 certificates, we begin with some useful background in Chapter 2. Section 2.1 introduces network scanning, Section 2.4 gives a brief background on the X.509 certificate and its corresponding infrastructure, and before continuing we state some important definitions in Section 2.10. The process of obtaining, parsing and working with the dataset from raw to analyzable data are further explored in Chapter 3. In Chapter 4, the results are presented and then discussed in Chapter 5. With fresh insight from this thesis, we round off with an outlook on similar and related research in Chapter 6. Finally, we conclude with a brief summary and some recommendations in the final Chapter 7.

2 Background

2.1 Network scanners

Network scanning is a technique used to probe all, or a subset of all, available Internet Protocol (IP) addresses (preferably in the IPv4 address space[1]) for specific ports or services using a specific protocol. This translates to sending individual requests to each and every IP address to see whether the receiving system (i.e. server) is configured to answer public requests made over a given protocol and/or port number used by the system. For instance, by initializing Transport Layer Security (TLS) handshakes over port 443, we can see which IPv4 addresses (and thus servers) that are open for Hypertext Transfer Protocol Secure (HTTPS) traffic. For research, so called responsible scanners are utilized that satisfies several conditions in order to be categorized as such [27].

The ZMap Project offers a collection of open-source tools that can be used for large-scale network scanning [21]. The creators of ZMap has analogously with the conditions for responsible scanners listed several criteria for what they call *"Good Internet citizenship"* that ZMap utilizes. ZMap has a capacity to traverse the complete IPv4 address space in less than five minutes, and less than 45 minutes with close too 100% hit rate [4].

2.2 Project Sonar

Project Sonar by Rapid7 is a research project based on network scanning, targeting over 70 services and protocols such as SSH (Secure Shell), SSL (Secure Sockets Layer) and DNS (Domain Name System) among others [43]. The data is made publicly available and is also used by Rapid7 to publish threat and exposure reports. The scanner is based on the above-mentioned ZMap, extended to support all studies being made by Rapid7 [47]. The specific study of interest in this paper is the SSL study. It has been active since 2013 and scans the complete IPv4 address space on a biweekly basis (weekly before mid-2015), omitting addresses listed in their continuously updated IP blacklist. Originally only focusing on the default HTTPS port 443, the project has grown over time to include an additional 37 non-443 ports with HTTPS endpoints as of April 2020 [50]. In addition to port numbers, the dataset contains

[1] As the total number of IPv4 addresses are the manageable amount of $2^{32} \approx 4.3 \cdot 10^9$ in total compared to the staggering $2^{128} \approx 3.4 \cdot 10^{38}$ possible addresses of IPv6.

IP addresses and unique so called SHA-1 (Secure Hash Algorithm 1) fingerprints of certificates, as well as incremental certificate files containing the full certificate data only of certificates identified for the first time. Worth noting is the fact that because the scan is IP based, in contrast to domain based, there is a small but possible chance of including ”accidental” certificate exposure not intended for usage – even though SSL/TLS has to be configured on port 443 in order for this to happen.

2.3 HTTPS and SSL/TLS

Originally, internet browsing was more or less exclusively made with Hypertext Transfer Protocol (HTTP) [41] as the application layer protocol. Over time, this has transitioned to its more secure sibling, Hypertext Transfer Protocol Secure (HTTPS) [44]. HTTPS utilizes encryption in the transport layer, namely Transport Layer Security (TLS) [45] or its now deprecated predecessor Secure Sockets Layer (SSL) [23].

As defined in its documentation, the default port number for HTTPS communication with HTTP/TLS over TCP/IP is port 443, even though other means of transport or other port numbers are allowed [44].

During the so-called handshake process initializing the TLS connection, the server sends the available certificates (see Section 2.4) to the client if certificates for authentication is determined as key exchange method [45]. This is how the Project Sonar SSL study is conducted, by probing all IP addresses with TLS handshakes requiring certificate authentication [43].

2.4 X.509 certificates

The certificates used in TLS are certificates in the Public Key Infrastructure (PKI) X.509 [11]. The standard defines everything from certification paths and trust, to operational and management protocols as well as revocations. Certificates are issued by specific certification authorities (see Section 2.5), whom are responsible for providing information regarding revocation status of the certificates issued through services such as Certificate Revocation Lists (CRLs) or Online Certificate Status Protocol (OCSP) (see Section 2.9).

Certificates contains a lot of standardized fields, and for the basis of this study some fields are of higher importance: signature algorithm, issuer, validity and subject. Some field are further divided into sub-fields, such as the issuer field defined as the X.501 type Name containing information about country, organization, etc.

A common certificate file format is the Privacy-Enhanced Mail (PEM) [35], which interestingly is not at all limited to email even though the name suggests so. Its usage with X.509 was further standardized in 2015 [32] and suggests Base64-encoded [31] data of certificates encoded with Basic Encoding Rules (BER) or Distinguished Encoding Rules (DER), defined in the X.690 standard [54]. The certificates in the Project Sonar SSL study is contained in PEM format.

2.5 Certification Authorities

Certification Authorities (CAs) are defined in the first version of X.509 as *”an authority trusted by one or more users to create and assign certificates”* [33, p. 20]. Whereas anyone can create their own CA, more guidelines are needed in order to earn public trust. The Certification Authority Browser Forum (CA/Browser Forum), maintains Baseline Requirements (BR) for the Issuance and Management of Publicly-Trusted Certificates [9] that addresses this trust requirement.

Certificates can also be issued by a CA to be used by themselves. Those certificates, being called ”self-signed”, have some commonalities different from the general characteristics of ”normal” certificates [24], one being validity time (see Section 2.8).

2.6 Automated CAs: Let’s Encrypt and cPanel

In the context of Certification Authorities, Let’s Encrypt is a CA of special interest in general and for this thesis in particular. By the number of certificates issued, Let’s Encrypt is by far the largest issuing CA as of last year [2]. The main difference between Let’s Encrypt and other CAs in general is that it is free, has support for automated certificate replacement and provides certificates with a validity period (see Section 2.7) of only 90 days – less than one tenth of the required maximum limit.

One similar CA is cPanel, providing 10% of all observed certificates, or 21% of all certificates for fully qualified domain names (FQDNs). Both Let’s Encrypt and cPanel automates some of their certificate services with the recently standardized Automatic Certificate Management Environment (ACME) [8]. Let’s Encrypt, for instance, has created its own automation tool Certbot as an ACME agent [51]. Of the most popular ACME clients, cPanel and Let’s Encrypt (Certbot) tops the list with a combined 33.9% of all issued certificates [2]. Worth noting is that cPanel system also issues other certificates, and almost one out of five Let’s Encrypt certificates are issued using cPanel [2].

Recently, a bug in the Let’s Encrypt issuing software lead to a massive revocation event in the beginning of March 2020 [28, 29], however for certain reasons it was not fully carried out [1, 46].

Given the above background, studying Let’s Encrypt and cPanel separately can be of certain interest going forward – especially regarding the mass revocation event in recent data.

2.7 Different certificate validations

Within the X.509 certificates, there are extensions aiming to provide the possibility of extended information such as community adaptions or agreements defined between CAs [11]. Some of these extensions must be recognized within the X.509 v3 standard, and one such extension is the *policy constraints* extension that tells which policy is used for issuing certificates. The CA/Browser Forum has listed Certificate Policy Identifiers, identified through Object Identifiers (OIDs), to be used with this extension in their Baseline Requirements (BR) and guidelines for Extended Validation (EV) [9, 26]:

`Domain Validated (DV):`	`(2.23.140.1.2.1)`
`Organization Validated (OV):`	`(2.23.140.1.2.2)`
`Individual Validated (IV):`	`(2.23.140.1.2.3)`
`Extended Validation (EV):`	`(2.23.140.1.1)`

The first three are defined in the CA/Browser Forum BR. The most basic of these validations is DV, which only requires proof of being in control of the domain name or the IP address of server in order to get a certificate issued. The remaining two validations of BR, namely OV and IV, requires the organization or individual to have their names validated in according to the requirements set up by the BR [9].

The last one, EV, is defined in the CA/Browser Forum Guidelines for the Issuance and Management of Extended Validation Certificates [26]. This is simply put an extended validation requirement where the CA verifies much more of the data provided in the certificate and based on this gives warranties of the information contained to be true.

The CA/Browser Forum BR allows Wildcard Domain Validation, meaning that the certificate holder can validate all subdomains of a domain as `*.domain.com`, where * is the wildcard character, if they can prove to be in control of everything under the domain level [9].

2.8 Certificate validity period

The validity of X.509 certificates are contained within the Validity field, consisting of the two timestamps `notBefore` and `notAfter` [11]. This validity period translates to the inclusive time when the CA guarantees to maintain certificate status data, such as in the event of a revocation. If at any time the certificate is either outside these validity boundaries or for some other reason cannot be validated (in the event of a revocation, for instance), the certificate is considered invalid.

Certificates following the CA/Browser Forum BR issued after 1 March 2018 are not allowed to be valid for more than 825 days (≈ 2 years + 3 months), and certificates issued before 1 March 2018 but no earlier than 1 July 2016 are not allowed to be valid for more than 39 months (≈ 1187 days, or 3 years + 3 months) [9]. EV certificates are required by the EV guidelines to not exceed 825 days, but a maximum of 12 months is recommended [26].

Looking at a 2016 study [13] also utilizing the Rapid7 Project Sonar dataset, valid certificates have a median validity of 1.1 years and a 90th percentile of 3.1 years – thus fitting well within the BR limits. Let's Encrypt pioneers within this area, issuing DV certificates with a validity of 90 days – less than 11% of the requirement by the CA/Browser Forum BR [2]

A 2019 study showed that 28% of certificates were self-signed and tended to have very long validity time. The most common validity periods were within the intervals 10-11 years and 22-23 years, and validity times up to 23 years ”only” constitutes 95% of all self-signed certificates [24].

2.9 Certificate revocations

During the validity of the certificate, the issuing CA must provide information regarding the certificate's revocation status. How this status is provided can be determined by the CA, but the documentation of X.509 lists Online Certificate Status Protocol (OCSP) and Certificate Revocation Lists (CRLs) as possible ways of doing it [11]. The latter, CRL, is included as its own section within the X.509 documentation, whereas OCSP is defined separately [25].

According to a 2015 study [36], not a single native mobile browser (on iOS, Android and Windows Phone) checks for revocation status of certificates, thus proving the current systems for revocations seriously unreliable.

2.10 Definitions

Birth: When a certificate is observed for the first time.

Death: When a certificate is observed for the last time.

Lifetime: The inclusive time between the first scan where a certificate is observed to the last scan it is observed [13]. (This is as such unrelated to the `notBefore` and `notAfter` values in the certificate, and only based on observations. In a 2016 study [13], the median lifetime of valid certificates is found to be 274 days.)

Onset period (X): The period between a certificates `notBefore` date and Birth.

Validity overlap (Y): The period between the new certificates Birth and the old certificates `notAfter`. Positive period is important to guarantee availability of service. Negative period will result in unavailability or a non-secure period.

Risk tendency: In the light of overlapping, the time from when the replacing certificate is allowed to be used for the first time and the time at which the old certificate expires. A greater overlap is regarded as a lower risk level and shorter overlap a higher risk level.

Replacement: A relation between a pair of certificates, where the IP address, port number and `subjectCN` matches, thus valid for the same entity and usage.

3 Method

In this chapter, the goal is to describe the journey from a raw dataset, explain the intermediate steps possible and taken, and finally reach a result that can be analyzed in terms of certificate replacements and general trends.

3.1 Retrieving the dataset

The Project Sonar SSL Certificates dataset is publicly available from Rapid7 [50] and while a few recently updated datasets are available without signing up, the complete collection is just a sign up away. Downloads are currently limited to 30 files per day, but can be increased by contacting Rapid7[1]. To speed things up, only the `_hosts` and `_certs` files for each date can be downloaded, as they together with their filename (stating the port number) contain all information required. For this study, only data for port 443 were used, which however constitutes the vast majority of all data in both absolute numbers and total file size. Data between 2017-04-25 and 2020-04-22 have been processed and analyzed, which sums up to 310 gigabytes of data.

The dataset is provided in `gzip` compressed text files. The `_hosts` files contains all responding IP addresses as well as the SHA-1 fingerprints of the certificates offered at each IP, with one IP-certificate pair per row. The `_certs` files contains only certificates observed for the first time, providing the full PEM encoded certificate as well as the aforementioned SHA-1 fingerprint. Thus, the complete picture is given by combining the data linked to the same SHA-1 fingerprints in both files.

3.2 Extracting the data

The data was parsed and processed in four steps, with the result of each step being saved to semicolon separated `.txt` files. If at any point the number of attributes for any given object did not match the expected number of attributes, the object was omitted.

[1] Hat-tip to the research team at Rapid7 for increasing our download limit to help with this thesis.

Certificate data	
Certificate	SHA-1 hash
	Serial number
Issuer / Subject	`CN`: Common name
	`OU`: Organizational unit
	`O`: Organization
	`L`: Locality
	`S`: State or Province name
	`C`: Country name
Validity	`notBefore`
	`notAfter`
Key info	Algorithm
	Public key length
Version	X.509 version
CA status	Subject is CA?

Table 3.1: Data extracted in Step 1

Step 1: Parsing the X.509 certificates

In the first step the certificate files were parsed using a Node.js library called *node-forge*[2]. With the help from this library, the data in Table 3.1 were extracted from the certificates and saved in a new file. From this step, close to 205 million certificates were extracted (see Appendix A.2).

Step 2: Calculating *birth* and *death*

In the second step, the birth and death dates of every certificate was calculated, recalling that birth is considered the first time a certificate was observed and death the last time it was observed in the hosts logs (see Section 2.10 for definitions). The updated logs were stored based on each certificates date of birth.

Step 3: Data duplication

In the third step the logs from step 2 were duplicated to be stored based on each certificates date of death. This was done to improve the execution speed of step 4.

Step 4: Calculating replacement relations

In step four, every adjacent pair of death-followed-by-birth log files were analyzed to find which dead certificate a new certificate replaced, if any. In order for a certificate pair to be considered a replacement pair (recalling the definition), there must be an exact match of IP address and port number between them, as well as an exact match of `subjectCN`. When a replacement pair was found, it was saved on the date of death of the dead certificate. Table 3.2 shows the content of a replacement relation log, and the relations are further explained in Figure 3.1. The total number of relations can be found in Appendix A.2.

[2] Available at: `https://www.npmjs.com/package/node-forge/v/0.9.0`

Relation data	
First certificate	SHA-1 hash
Log data	IP address
	Port number
Issuer/Subject	`CN`: Common name
	`C`: Country name
Key info	Public key length
Validity	`notBefore`
	`notAfter`
Lifetime	*Birth*
	Death
Calculations	x: Onset period
	y: Overlapping period
CA status	Subject is CA?
Replacing certificate	Same Issuer?
	SHA-1 hash

Table 3.2: Data compiled in Step 4

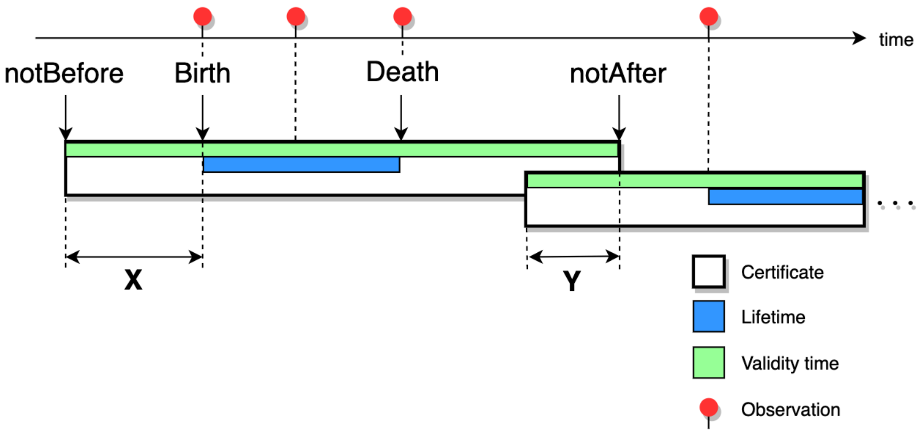


Figure 3.1: Replacement relation between two certificates, where X and Y are onset and overlapping period respectively (see Section 2.10).

3.3 Analyzing the replacement relations

Two different analyzes were made, both with regards to the certificate overlap. The first is the distribution of overlapping days and the second the distribution of overlap relative to the validity time of the certificate. The data was aggregated based on the following:

- Continent for Issuer country (Issuer's `C` value)
- Continent for Subject country (Subject's `C` value)
- Validity time intervals
- The ten most commonly used Issuers according to a 2019 study [2] (see Table 3.3)

A country information file[3] (corrected with Puerto Rico belonging to North America rather than ”America”) was used to get the continent from a two-letter country code. If an issuer or subject did not provide a two-letter country code present in this list, that certificate was omitted from the result.

[3] By Pamela Fox and Misha Lisovyi. Available at:
`https://gist.github.com/mlisovyi/e8df5c907a8250e14cc1e5933ed53ffd`

Most popular CAs	
Let's Encrypt	57%
cPanel	21%
Sectigo	7%
DigiCert	4%
Cloudflare	3%
GoDaddy	2%
GlobalSign	0.9%
Nazwa.pl	0.5%
Amazon	0.4%
Starfield	0.4%
TrustAsia	0.2%
Other	*1.7%*

Table 3.3: Most popular CAs [2]

3.4 Limitations

The Project Sonar studies normally perform scans on a biweekly basis [50], thus having a granularity in the resulting dataset with no possibility to analyze what happens in between. This is suitable for long-term trends, but not as usable for details on smaller time units.

The chosen solution for this study was time consuming, as several steps increased linearly in time thus limiting the interval of data studied. As large data had to be processed, I/O speed limited the possible computation speed the most. This could potentially be improved by utilizing more I/O units to store files and combined with larger working memory size and multiple threads for computations this can be greatly improved.

As the ideal maximum validity period is 3.25 years (see Section 2.8), a dataset with less data than this maximum period is clearly not ideal. The initial data for certificates of shorter validity will be the most accurate, but as the validity period grows and/or the birth gets closer to the final period of observation, the reliability of the results gets limited. There are however many shorter certificates, which allows for some conclusions to be made.

4 Results

In this chapter, the results found by processing the dataset will be presented, giving some general reflections before diving into some specific highlights. To guide in the interpretation, a brief introduction to the results are given first.

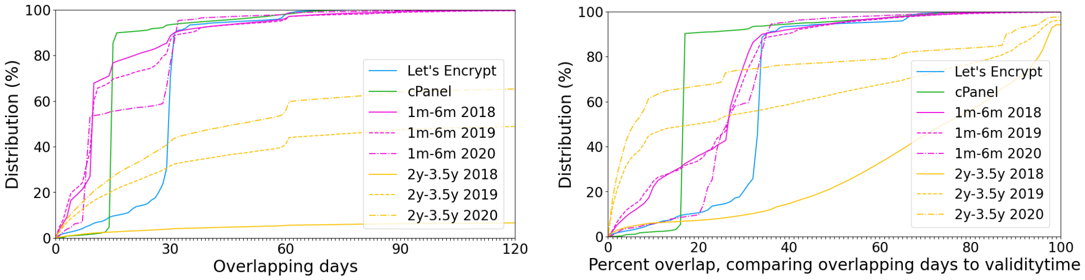
Interpreting the graphs and tables

The graphs presented in this section shows the distribution of certificates in different categories, presented as CDFs, cumulative distribution functions. The value on the y-axis show the percent count of certificates in each category that falls under the threshold shown on the x-axis. When looking at the graphs, a perfect plateau means that nothing has changed at that moment and on the contrary the steeper the slope at any given x value, the more certificates are registered at that exact threshold.

To show longitudinal data, whisker plots are used presented in order of appearance starting from the bottom. The bottom horizontal line of a whisker plot shows the smallest value, the bottom of the box show the threshold for the 25th percentile, the red line the threshold for the 50th percentile (median), the top of the box show the threshold for the 75th percentile and finally the horizontal line at the top indicates the highest recorded value.

Some general notes on the results

When grouping the certificates based on whether the replacing certificate was issued by the same issuer or another, the overlap was shorter when a subject switched issuer compared to when using the same (see Appendix A.1). Server administrators do not seem keen on switching certificates quickly when receiving a new, but rather uses a certificate for the better part of its validity time. The cost of certificates might also be a contributing factor to using certificates for as long as possible before renewing. This same-or-other issuer difference could be explained by the fact that Let's Encrypt probably accounts for most of the certificates being replaced by the same issuer. The graphed data is based on all certificates, and the use of Let's Encrypt certificates show a strong connection to a 30-day overlap policy supporting this idea. Nonetheless, the result stands true and the strong compliance with a 30-day overlap might be one of the reasons why Let's Encrypt certificates are so widely used. The behavior of certificates issued by Let's Encrypt can similarly be found in certificates issued by cPanel, and



(a) Overlapping days

(b) Overlap relative to validity time

Figure 4.1: Certificate overlap distribution highlights

Group	Percent	Count
Let's Encrypt	73.49%	11 447 676
cPanel	11.37%	1 771 297
1m-6m 2018	0.46%	71 420
1m-6m 2019	0.26%	40 991
1m-6m 2020	0.17%	26 689
2y-3.5y 2018	0.72%	112 804
2y-3.5y 2019	1.03%	160 197
2y-3.5y 2020	0.77%	120 206

Table 4.1: Certificate count data for overlap highlights figure

as mentioned in the background, a significant amount of Let's Encrypt certificates are issued through cPanel.

When grouping the certificates based on validity time there is a 5-10 percentage points spike on 60 days of overlap for certificates with a validity time of 6 months and above (see Appendix A.1). This could be caused by three of the selected top-10 issuers, namely Amazon, GoDaddy and Starfield. Looking at Figure 4.2 (and Figure A.6 in Appendix), they all show a spike at 60 days overlap when looking at certificates they have issued.

4.1 Variance

Validity time

The first thing to note about Figure 4.1 is that the reason for the rapid decline in overlapping days regarding the 2 years to 3.5 years interval (2y-3.5y) is caused by the method, as only early replacements having a longer overlap would be caught in the 2018 summary.

Another thing to note about the same figure is that certificates with longer validity time have a more even spread of overlap distribution relative to certificates with shorter lifetime. This might suggest that the overlap of short-validity certificates is influenced by the Issuer whilst long-validity certificates are influenced by the subject to a greater extent.

Issuer Common Name

It is obvious that the subject has influence over the certificate overlap as they can simply get a new certificate from one of many issuers at any point in time, given that only domain validation is required. But how much impact does the issuer have regarding the overlap? From Table 4.3 we can conclude that certificates from cPanel and Amazon show the most desirable statistics with over 98% of their certificates having an overlap when being replaced and DigiCert, Let's Encrypt and GlobalSign close behind with more than 90% of their issued certificates having

	2018		2019		2020 (to April)	
Group	**%**	**Count**	**%**	**Count**	**%**	**Count**
Let's Encrypt	81.31%	4 746 074	82.96%	5 320 692	83.15%	1 380 910
cPanel	15.62%	911 513	11.23%	719 912	8.42%	139 872
Sectigo	0.00%	0	0.13%	8 602	1.63%	26 995
DigiCert	1.08%	62 941	1.82%	116 620	3.06%	50 801
GoDaddy	1.03%	60 113	1.51%	96 809	1.90%	31 543
GlobalSign	0.49%	28 338	0.73%	46 909	0.80%	13 275
Nazwa	0.00%	239	0.01%	372	0.01%	87
Amazon	0.18%	10 333	1.09%	69 711	0.48%	7 921
Starfield	0.20%	11 640	0.29%	18 396	0.36%	6 056
TrustAsia	0.10%	5 767	0.24%	15 269	0.19%	3 216

Table 4.2: Number of certificate, grouped by issuer, over time

Group	# with overlap	# with gap	% with overlap
Let's Encrypt	11 447 676	717 431	94.10%
cPanel	1 771 297	21 137	98.82%
Sectigo	35 597	9 749	78.50%
DigiCert	230 362	10 726	95.55%
GoDaddy	188 465	43 548	81.23%
GlobalSign	88 522	7 104	92.57%
Nazwa	698	434	61.66%
Amazon	87 965	1 054	98.82%
Starfield	36 092	7 164	83.44%
TrustAsia	24 252	10 319	70.15%

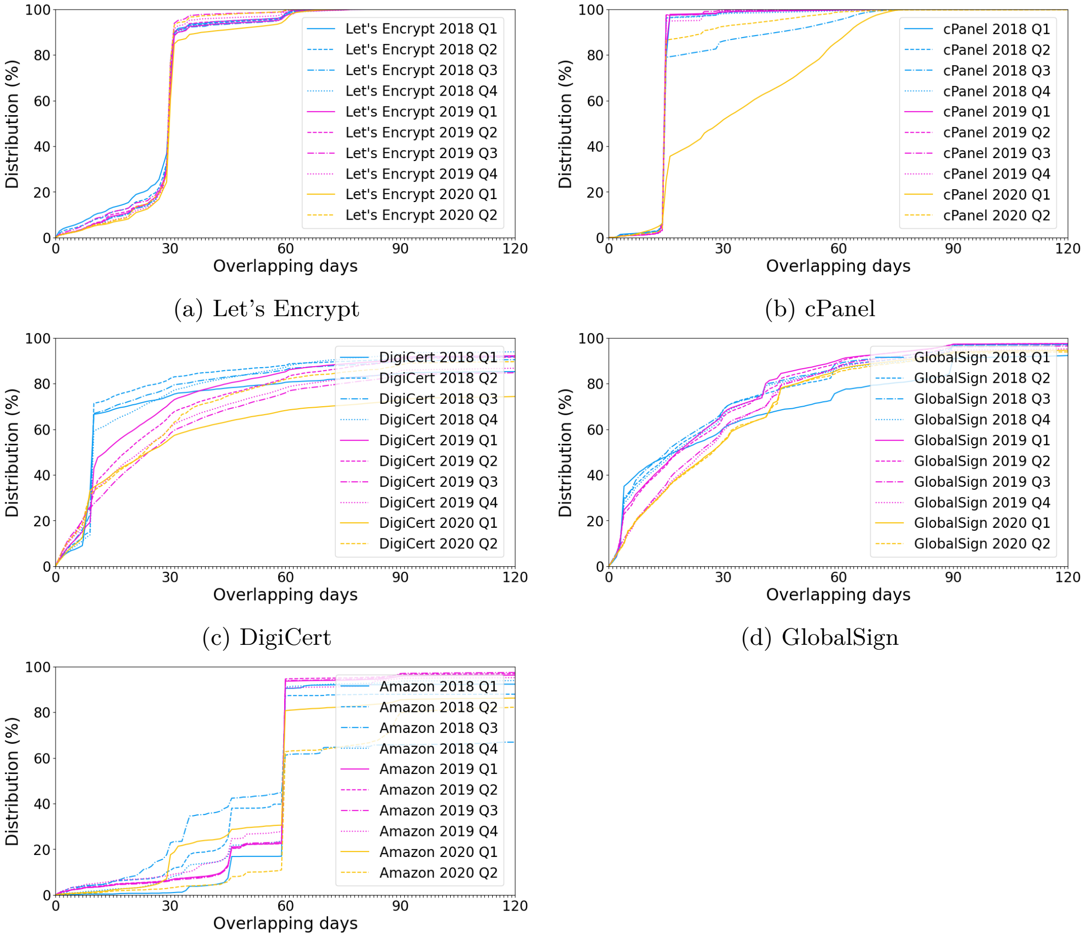
Table 4.3: Number of certificates with and without overlap, grouped by issuer

an overlap when being replaced. Is this just a matter of coincidence or does the issuers play a part in this?

Consider an exponential growth to be caused by strictly managed update policy, and a logarithmic growth to be caused by the subject taking the initiative for the renewal of a certificate. Furthermore, consider the height of a spike the reach of a policy and the width how strict the policy is. Figure 4.2 shows how the five top performing issuers mentioned above have performed on a quarterly basis from January 2018 to April 2020. From the look of these graphs, both Let's Encrypt and cPanel seem to play a major part in their users' certificate renewal process. The same can be said about Amazon, especially considering 2019, and DigiCert during 2018. Moreover, it seems to be the case that the responsibility regarding certificate renewal is put on the subjects for certificates from GlobalSign, and DigiCert after 2018.

The graphs shown in Figure 4.2a and 4.2b supposedly shows the impact of automated certificate renewal systems, and how well they can direct users to adhere to a specified user behavior. Let's Encrypt seem to promote a 30-day overlap, reaching 70% of its user base, and cPanel a 15-day overlap, reaching 95% of its user base with this policy.

The reason for the different characteristic of the cPanel 2020 Q2 line can be found by looking at Figure 4.3b and Table 4.4 together with the knowledge that cPanel administrates Let's Encrypt certificates for its customers. With the numbers from the table we can conclude that more of Let's Encrypt's certificates were affected by their mass revocation event than cPanel's, and with the figure that a higher degree of cPanel certificates were affected. The later might be a precaution by cPanel simply deciding to update all their issued certificates for their clients to minimize their overhead and still be on the safe side.



(a) Let's Encrypt

(b) cPanel

(c) DigiCert

(d) GlobalSign

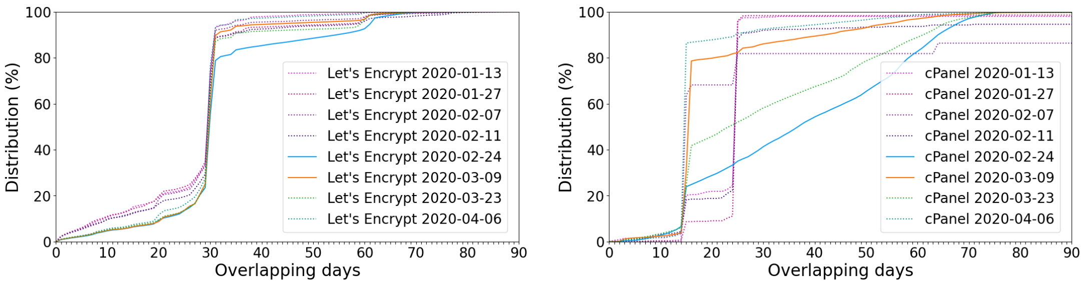
(e) Amazon

Figure 4.2: Certificate overlap distribution of top performing issuers on a quarterly basis

Figure 4.3 shows the impact of Let's Encrypts mass revocation event 2020-03-05. The most interesting data to look at are the two solid lines, 2020-02-24 and 2020-03-09. The first of the solid lines catches the overlap from certificate replacements including those triggered somewhere around 2020-03-05 as a result of the revocations, thus making the replacing certificates observed at the following data point, 2020-03-09. The second solid line shows the overlap distribution for certificates who died 4 to 18 days *after* the revocation event (after 2020-03-09 and before 2020-03-23 to be precise). A significant change in any of these lines' characteristics would indicate the effect the event has had on certificate replacement behavior.

To see the aftermath of the revocation event, and if this has triggered a fluctuating replacement intensity, one would need data up until May 2020 as this would be the time when most of the newly created certificates would have been replaced. Including May would further include all certificates affected by the revocation event as they would be past their validity (due to their 90-day lifetime). However, looking only at the available data, the graphs in Figure 4.3 shows the revocation seem to have disrupted the replacement flow of about 10% of certificates from Let's Encrypt being replaced directly after the revocation event, and 70% of certificates from cPanel being replaced directly after the event, showing longer lasting effects for cPanel but not for Let's Encrypt.

When instead looking at the numbers behind the graphs, shown in Table 4.4, the event proves to have had a significant impact on both Let's Encrypt and cPanel. The number of observed certificate replacements, replacing a Let's Encrypt certificate with a new one (same or



(a) Let's Encrypt

(b) cPanel

Figure 4.3: Impact of Let's Encrypts mass revocation event 2020-03-05

	Let's Encrypt		**cPanel**	
Date	**Percent**	**Count**	**Percent**	**Count**
2020-01-13	0.52%	7 201	0.25%	352
2020-01-27	0.45%	6 225	0.56%	782
2020-02-07	0.19%	2 690	0.02%	22
2020-02-11	0.64%	8 892	0.22%	314
2020-02-24	24.81%	342 552	51.86%	72 539
2020-03-09	19.52%	269 599	11.15%	15 600
2020-03-23	20.15%	278 295	15.94%	22 298
2020-04-06	33.71%	465 456	19.99%	27 965

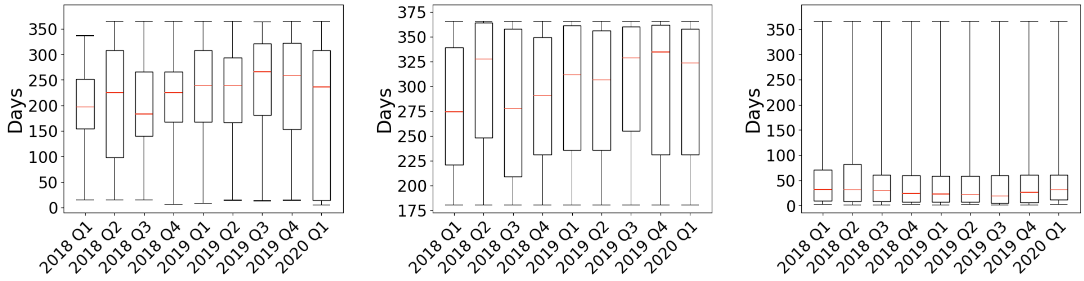
Table 4.4: Impact of Let's Encrypts mass revocation event 2020-03-05 in numbers

other issuer), increased with over 3 750% from 2020-02-11 to 2020-02-24 (the first date showing the impact of the mass revocation) and regarding cPanel the number of observed replacement increased with a whopping 23 000% regarding the same interval. Both issuers have a significant increase in observed certificate replacements until the end of the analyzed time frame of this study. This might be caused by more users integrating Let's Encrypt's/cPanel's auto renewal systems with their platform. It could also be the result of users switching away from Let's Encrypt certificates – or something entirely else. As this is just speculations, this is suggested to be studied further in Section 7.1.

4.2 Trends

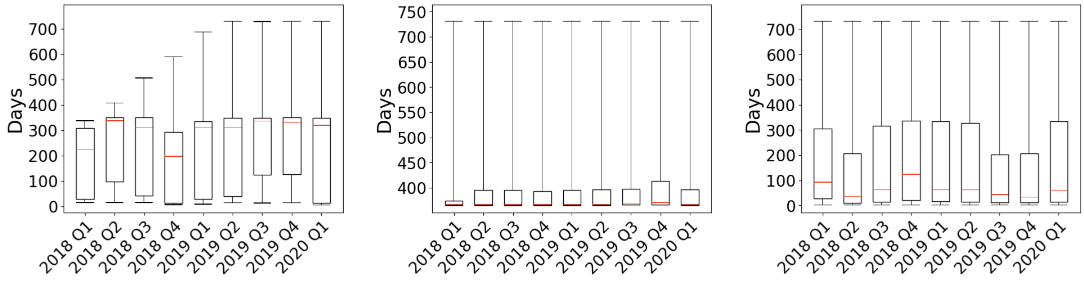
Insights about certificates with a validity time above 1 year must be done with caution as good utilization of their validity time would mean more certificates showing up towards the end of the observed interval. This can clearly be seen in Figure 4.6a. It is clear that the top values are affected by this, but it is also true for the bottom as more long-lived certificates can change the threshold for the 25th, 50th and 75th percentiles in a quarter. Nonetheless, a few things can be said about this result. Looking at Figure 4.6 the overlap decreases over time. The high initial levels are probably caused by revoked long-lived certificates, resulting in a long overlap. The data at the end is probably more accurate as planned updates of long-lived certificates would not appear at the start of the visualized data set. Replacement of 2-year certificates with good utilization of validity time would first appear in the dataset at Q2 2019, as seen in Figure 4.5b. From this point, the result is skewed in favor of 2-year certificates and the 3-year certificates are not represented in a comparable way.

The aforementioned 2-year certificates are represented from Q2 2019 and can be seen in Figure 4.5a. Looking at the same figure, this raises the question why the bottom 25th percentile is significant higher for Q3 and Q4 2019 compared to Q2 2019 and Q1 2020. The reason for this is unknown but could be the result of a break point close to the 25th percentile tipping



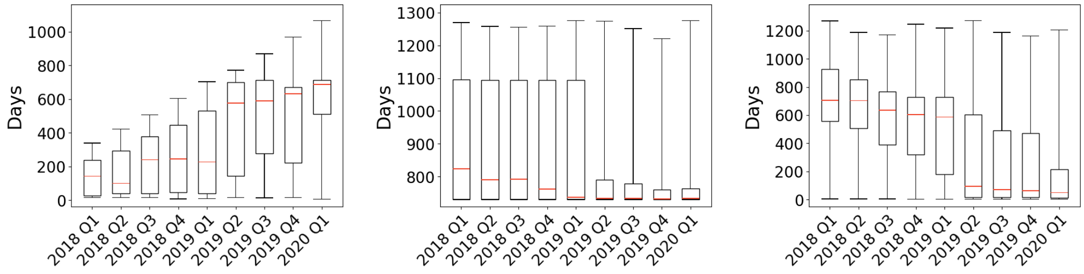
(a) Lifetime (b) Validity time (c) Overlap

Figure 4.4: Certificates with validity time between 6 months and 1 year



(a) Lifetime (b) Validity time (c) Overlap

Figure 4.5: Certificates with validity time between 1 and 2 years



(a) Lifetime (b) Validity time (c) Overlap

Figure 4.6: Certificates with validity time between 2 and 3.5 years

back and forth as more certificates with a validity time of 2 years are being replaced. Worth noting is that over 25 percent of the certificates have a lifetime of less than 25 days, despite having a validity time of at least 365 days. This suggests a bad usage behavior as 25 percent of the certificates are not likely to have been revoked.

5 Discussion

5.1 Results

There are many aspects to security concerning X.509 certificates. The overlap is important to guarantee the availability of the service that the certificate protects. Another important aspect is that the validity time should not exceed the actual use time, as it is then not considered secure as status data is not maintained by the issuing CA. As X.509 private keys potentially could be found with brute force, keeping the validity time low enough will make this type of attack harder to pull off. The current CA/Browser EV Guidelines [26] for instance recommends that EV certificates should be valid for a maximum of one year. In order to find a balance between what the certificate should be used for and keeping it secure the entire time, the certificate should have a fairly short validity time however still leaving room for some flexibility in terms of certificate replacement period.

The internet security landscape is up for a big change. Due to a heavy growth of machine-to-machine (M2M) communication, to a large extent explained by Internet of Things (IoT), new M2M connections are expected to more than double over a five year period [15], raising questions regarding whether the security protocols of today meet our upcoming security needs. Mobile devices are, as mentioned in the background, known to not verify whether a X.509 certificate has been revoked or not, effectively rendering this security measure useless. A logical reason for this is to save battery and expanding on this thought one might guess that IoT devices will also follow this pattern of not verifying whether an X.509 certificate has been revoked or not. With this said, X.509 certificates need to be secure by design: *Always secure.*

Always secure, by design

Based on the CA/Browser Forum Baseline Requirements, a corrupt certificate normally must be revoked within 5 days from the point of discovery [9]. This raises some questions, for instance how soon a compromised or in any other way corrupt certificate would be discovered, and what type of damage that could potentially already have done. And how could this affect mobile devices long-term? In a ”best worst case” scenario (an immediate discovery, but certificate is revoked just before the required deadline), an attacker could compromise your system for 5 days doing who knows what damage. And this only covers desktop clients, still leaving mobile clients vulnerable until the certificate’s `notAfter` date. With *always secure*, a certificate's maximum validity time would be capped at 5 days, resulting in the security of a service never performing worse than the ”best worst case” scenario independent of the client's device. This could of course be even shorter in order to be safer, but 5 days is reasonable given that it complies with industry agreements.

Let's Encrypt and cPanel have shown that it is both possible to decide on a desired certificate overlap and actually achieving it, thus never putting the availability at risk. With that said, the tools are already in place to adopt the *always secure* policy in order to strengthen internet security.

5.2 Method

In terms of replicability, the study is deterministic in the sense that following the method presented should render the same exact result. The dataset is provided by Rapid7, and ”cannot” retroactively be updated as it is a snapshot of the time the study was conducted.

Reliability

As the used method is quantitative with a static dataset, the reliability in the result is determined based on the quality of the data and the data loss rate during processing and analyzing. Regarding the quality of the dataset, there are primarily three pitfalls that could affect the quality of the dataset: The number of servers who have actively opted out from allowing crawlers, the number of servers being down during the time of scanning and lastly the possibility of networking issues. The ZMap project, which Project Sonar is based on, provide a 100% hit rate when operating at slower speeds [4], thus meaning that the networking issues are negligible, and limitations due to opting out are unavoidable in order to conduct ethically valid studies.

Errors resulting from temporarily unavailable host should be limited as the presented result are aggregated on a yearly basis, thus these losses can be considered evenly spread out affecting different groups of certificates proportionally.

As the scans are conducted on a rough biweekly basis, this means that certificate replacement relations will most likely be incorrect in terms of short-lived certificates: For instance, certificates replaced daily would have had 13 undetected intermediate certificates in between the scans.

Validity

Sanity checks of the result can be done by analyzing the trend graphs (see Appendix A.1). The first thing to note is that the average validity time for certificates issued by Let's Encrypt is a perfect line at 90 days validity the same length as all certificates Let's Encrypt provide. This shows that the processing of timestamps works as expected.

Limitations on possible insights

Considering comparing trends on a yearly or quarterly basis, the results from the earlier years in the data set are overrepresented by short-lived certificates, and only the latter data gives a more truthful picture. This is because a certificate is required to be both observed and die within the observed time frame. It also must be replaced by a new certificate in order to be stored as a replacement relation and it is the replacement relations that constitutes the source for the end-analysis of certificate usage behavior. This fact becomes evident when looking at Figure 4.6a, and Figure 4.5a to some degree as well.

The limitation of only having data points from every other week means no real conclusion regarding overlap can be derived from certificates with shorter validity times: especially below two weeks, but also up to one month (however not as significant).

Corrupt data implications

When parsing the data, all certificates where any of the stored fields included a semi-colon would not be included in the data analyzed, as this character was used as a delimiter between fields. If a field were to include this character, the following step would render some following issues due to the semi-colon being central in delimiting the data and was therefore discarded. The omission of these certificates is however considered to be non-crucial, as on average they account for less than 0.7% of the certificates in any given log file. Worth noting is the corrupt certificate file from 2018-01-30 primarily resulting in more intermediate certificates being missed between the two certificates in a replacement relation. The impact of this is considered to be evenly distributed across the rest of the logs but would have a greater impact within close proximity. As the data is aggregated to quarterly reports, the impact of this one missing file is considered negligible and as such the final dataset is considered to be reliable.

On a per-quarter basis, the corrupt file from 2018-01-30 has a bigger and unknown impact, especially on the first quarter from 2018. The impact on the following results however should not be that much, and these certificates are omitted from the death logs. The birth of certificates not being recognized means that replacement relations for the closest foregoing timestamp cannot be created and therefore not included in the analysis. These certificates are also omitted from the death dataset of every following deaths log, skewing the aggregated results of replacement relation logs. Even so, this is not a major problem as the missing certificates are assumed to be evenly distributed.

Certificate replacement detection

When determining the birth and death of a certificate, the birth date as well as the IP address and port number is set based on the first observation. However, the death date is determined based on the last observation of the certificate. If the certificate is being used on multiple address/port combinations, and the certificate is replaced on one of them before it is replaced on another, the method used would either say that the certificate was never replaced or think it was replaced by a later certificate following it. It is fair to say that a certificate is dead only when it is no longer being used on any combination, but in order to get the replacement right, the IP address and port number should be updated to be the same as the server IP and port number where it was observed the last time.

Certificate updates partly changing the scope can potentially be missed, as the certificate replacement logic requires an exact match between the subject's common name of the two certificates in order to be considered a replacement. The problem occurs when for instance the old certificate includes a wildcard and the new certificate is more precise, i.e. a sub-domain. The same happens with the opposite, when the new certificate includes a wildcard and the old one does not. Implementing a matching function allowing wildcards in a subject's common name would most likely increase the number of certificate replacements detected and therefor improve the final result.

Project Sonar only stores observed certificates once, regardless of what port the scan was done on. A replacement relation could be missed as a result of this if said certificate is being used on multiple ports and was first observed on a non-443 port. This is due to the design that certificates are only included in the subset where it was first observed, thus potentially rendering some used certificates outside the scope of this analysis. However, this should not be considered a problem, as the number of certificates on non-443 ports are extremely few compared to port 443. It could be solved by including all available data, however largely increasing the number of files (but not total file size as much).

5.3 Certificate selection

Because self-signed certificates tend to have their own characteristics, including their extensive validity times, they might as well be omitted from the analyzed certificates. In the used method, this was done by adding a filter removing all certificates with a validity over 3.5 years (thus removing those not following the CA/Browser Forum Baseline Requirements). However, this could just as well – and possibly more effective – have been done by removing self-signed certificates early on processing the certificate data.

5.4 The work in a wider context

A note on ethical considerations

Scanning the entire IPv4 address spaces without prior consent of courses raises some ethical questions. As previously mentioned in connection to network scanning (see Section 2.1-2.2), these scanning techniques follow a set of transparent rules and a public disclosure of how it operates as well as how to opt-out [21]. In addition to this, the projects are open-source, performs randomized probing to avoid network overloading and have built in exclusion capabilities.

The resulting data contains only public data, in the sense that its open for everyone to collect by just initializing a TLS handshake to a given IP address. In addition to this, the main logs contain only IP addresses and port number, and as such only rarely can be used to identify any specific person. While the data of the returned certificate contains a lot of information, it's still non-personal and primarily linked to domains or organizations.

6 Related work

The basis for this thesis has been to use datasets based on network scanning. What is possible to do in this area is explored in a 2015 paper [18] introducing a query engine based on such scanning, and a 2017 study of cryptographic libraries [40]. The limitations of current systems when transitioning to IPv6 is addressed in a 2017 paper [39], and the performance with different protocols as well as the concept of ”liveness” is explored in a 2018 article [6].

Interestingly, not much research has been done in the field of certificate overlapping. Whereas this thesis focuses primarily on certificates being used and replaced within their predefined validity periods, current research and development revolve around the occurrence of certificate revocations.

One study shows that even though measurements are in place to revoke certificates, a non-negligible share of certificates identified through network scanning have been revoked [36]. The same study also points out the fact that built-in revocation checks in browsers are far from satisfying, especially mobile browsers that effectively never checks.

The issues with revocations was extensively studied as the aftermath of both DigiNotar [5] and Heartbleed [20, 55] where issues with revocations and replacements turned out painfully obvious. Thorough analysis of the HTTPS and certificate landscape around the time of DigiNotar and Heartbleed can be found in a 2011 [30] and 2013 [16] paper respectively. Naturally, one area of broad research interest is thus better ways of revoking certificates. While the goal is the same, the path taken differs widely with several different solutions available [14, 17, 34, 49]. Some initiatives, such as CRLite [34], addresses the fact that the current pull model does not seem effective, and instead proposes various forms of push models. Some initiatives revolves around new or evolved strategies, such as CRT [17], and some around extended certificate data, such as OCSP Must-Staple [14].

The usage of expired certificates is explored in some research. One study analyzes different root causes of certificate errors, finding that effectively all date errors on the server side are caused by expired certificates and that more than half of those expired within the last 30 days [3]. A previous study within the same area suggest that the problem with expired certificates are notably (3x) more common among self-signed certificates [19], and another study that as much as 88% of invalid certificates in general are self-signed [13].

Regarding usability of HTTPS deployment, a recent study [10] focused on its general non-friendliness and the fact that Let’s Encrypt is helping to improve it. Let’s Encrypts usability with its Certbot is further explored in another recent study [51]. The effect of this is then proved by numerous studies showing its impact in the long tail of non-top-1M sites [2, 12, 38].

Another perspective of certificates is what protocols are used in practice, in contrast to what could potentially be used by the involved systems. This is explored in a 2017 article, looking at cipher suits and the discrepancy between best practice and reality [42].

7 Conclusion

In this thesis, we have shown that the Project Sonar SSL dataset is suitable for making observations and finding insights regarding HTTPS certificate usage. Our main conclusion is therefore, summarizing all our main questions:

> **Certificate lifetime in relation to its validity depends heavily on issuer and the issued validity period. In general, shorter validity periods and automated issuing tends to result in a more constant lifetime and predictable replacement characteristic, which is very beneficial in terms of internet security.**

With this background, we propose an *always secure* policy where certificates are always issued with a validity of maximum 5 days, thus meeting the requirements for time-to-revocation stipulated by the CA/Browser Forum Baseline Requirements. This ”secure by default” solution effectively removes the need for costly revocations, and issuers such as Let’s Encrypt and cPanel have proven these automated solutions to be possible at scale. Using such automated certificate renewal greatly increases the security of your HTTPS service, allowing for shorter validity times without putting the availability of service at risk.

When selecting which CA to issue your next X.509 certificate, the specific CA you choose is of less importance than the way you and your organization use the certificates. However, all certificate authorities in the top tier for consideration *should* provide automated renewal features, as this is associated with the most secure certificate overlapping behavior observed in the dataset. If the chosen CA provide automated renewal of certificates, certificates with shorter validity periods are preferred as they provide better security by simply having a shorter validity period. To ensure the availability of your service, having too short validity might prove challenging. However, we still propose a maximum of 5 days in order to always stay secure and still allowing room for short periods of maintenance or downtime. As a recommendation to CAs who does not provide automated renewal features, providing this service should be seriously considered, as well as providing certificates with validity periods of 5 days.

7.1 Further studies

Given the exploratory nature of this thesis, many related questions for further studying surfaced. One interesting start would be to continue looking at HTTPS trends. The available

HTTPS dataset could be combined with an HTTP dataset to see what the HTTPS adoption trend looks like, and what drives it? Another perspective would be exploring the behavior using IPv6 addresses, either finding a completing dataset or launching own studies.

Expanding on the result of this thesis, further research in whether Q4 2019 marks a trend shift for certificates valid 2 to 3.5 years regarding validity would be interesting to investigate. One could also dive into the area of automatic certificate renewal systems, to see how they work and how this reflects in the extracted data. Is there an automation adoption trend, and if so, what does it look like?

A few other low-hanging fruits for further exploration are: Expanding this study with more X.509 headers to analyze, in other ways extending the analysis to improve the results and look at long-term effects of the Let's Encrypt mass revocation event. To find better results regarding short-validity-time certificates, it is tempting to analyze a dataset based on a more frequent study allowing for daily data instead of biweekly. Another really interesting study would be combining the dataset with revocation data to see the interplay between revocations and different classes of certificates. And all these ideas are just some of the ideas at the top of our heads, given the endless possibilities that these datasets provide. We hope some of them can inspire further research within this area.

Bibliography

[1] Josh Aas. *Let's Encrypt: Incomplete revocation for CAA rechecking bug.* Bugzilla. 2020. URL: https://bugzilla.mozilla.org/show_bug.cgi?id=1619179 (visited on 03/19/2020).

[2] Josh Aas, Richard Barnes, Benton Case, Zakir Durumeric, Peter Eckersley, Alan Flores-López, J Alex Halderman, Jacob Hoffman-Andrews, James Kasten, Eric Rescorla, and Et al. "Let's Encrypt: An Automated Certificate Authority to Encrypt the Entire Web". In: *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security.* CCS '19. London, United Kingdom, 2019, pp. 2473–2487. DOI: 10.1145/3319535.3363192.

[3] Mustafa Emre Acer, Emily Stark, Adrienne Porter Felt, Sascha Fahl, Radhika Bhargava, Bhanu Dev, Matt Braithwaite, Ryan Sleevi, and Parisa Tabriz. "Where the Wild Warnings Are: Root Causes of Chrome HTTPS Certificate Errors". In: *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security.* CCS '17. Dallas, TX, USA: Association for Computing Machinery, 2017, pp. 1407–1420. DOI: 10.1145/3133956.3134007.

[4] David Adrian, Zakir Durumeric, Gulshan Singh, and J Alex Halderman. "Zippier ZMap: Internet-Wide Scanning at 10 Gbps". In: *Proceedings of the USENIX Conference on Offensive Technologies.* WOOT '14. San Diego, CA, USA, Aug. 2014, p. 8. DOI: 10.5555/2671293.2671301.

[5] Johanna Amann, Oliver Gasser, Quirin Scheitle, Lexi Brent, Georg Carle, and Ralph Holz. "Mission Accomplished? HTTPS Security after Diginotar". In: *Proceedings of the Internet Measurement Conference.* IMC '17. London, United Kingdom, 2017, pp. 325–340. DOI: 10.1145/3131365.3131401.

[6] Shehar Bano, Philipp Richter, Mobin Javed, Srikanth Sundaresan, Zakir Durumeric, Steven J Murdoch, Richard Mortier, and Vern Paxson. "Scanning the Internet for Liveness". In: *SIGCOMM Comput. Commun. Rev.* 48.2 (May 2018), pp. 2–9. DOI: 10.1145/3213232.3213234.

[7] Richard Barnes. *Deprecating Non-Secure HTTP.* Mozilla Security Blog. 2015. URL: https://blog.mozilla.org/security/2015/04/30/deprecating-non-secure-http/.

[8] Richard Barnes, Jacob Hoffman-Andrews, Daniel McCarney, and James Kasten. *Automatic Certificate Management Environment (ACME)*. RFC 8555. RFC Editor, Mar. 2019. DOI: 10.17487/RFC8555.

[9] *Baseline Requirements for the Issuance and Management of Publicly-Trusted Certificates.* Version 1.7. CA/Browser Forum, 2020. URL: https://cabforum.org/wp-content/uploads/CA-Browser-Forum-BR-1.7.0.pdf.

[10] Matthew Bernhard, Jonathan Sharman, Claudia Ziegler Acemyan, Philip Kortum, Dan S Wallach, and J Alex Halderman. "On the Usability of HTTPS Deployment". In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. CHI '19. Glasgow, Scotland, UK, 2019, pp. 1–10. DOI: 10.1145/3290605.3300540.

[11] Sharon Boeyen, Stefan Santesson, Tim Polk, Russ Housley, Stephen Farrell, and Dave Cooper. *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*. RFC 5280. RFC Editor, May 2008. DOI: 10.17487/RFC5280.

[12] William J. Buchanan, Scott Helme, and Alan Woodward. "Analysis of the adoption of security headers in HTTP". In: *IET Information Security* 12.2 (2018), pp. 118–126. DOI: 10.1049/iet-ifs.2016.0621.

[13] Taejoong Chung, Yabing Liu, David Choffnes, Dave Levin, Bruce MacDowell Maggs, Alan Mislove, and Christo Wilson. "Measuring and Applying Invalid SSL Certificates: The Silent Majority". In: *Proceedings of the Internet Measurement Conference*. IMC '16. Santa Monica, CA, USA, 2016, pp. 527–541. DOI: 10.1145/2987443.2987454.

[14] Taejoong Chung, Jay Lok, Balakrishnan Chandrasekaran, David Choffnes, Dave Levin, Bruce M Maggs, Alan Mislove, John Rula, Nick Sullivan, and Christo Wilson. "Is the Web Ready for OCSP Must-Staple?" In: *Proceedings of the Internet Measurement Conference*. IMC '18. Boston, MA, USA: Association for Computing Machinery, 2018, pp. 105–118. DOI: 10.1145/3278532.3278543.

[15] *Cisco Annual Internet Report (2018–2023)*. Cisco Public White Paper. Cisco, 2020. URL: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.pdf.

[16] Jeremy Clark and Paul C. van Oorschot. "SoK: SSL and HTTPS: Revisiting Past Challenges and Evaluating Certificate Trust Model Enhancements". In: *IEEE Symposium on Security and Privacy*. SP '13. Berkeley, CA, USA, May 2013, pp. 511–525. DOI: 10.1109/SP.2013.41.

[17] Luke Dickinson, Trevor Smith, and Kent Seamons. "Leveraging Locality of Reference for Certificate Revocation". In: *Proceedings of the Annual Computer Security Applications Conference*. ACSAC '19. San Juan, Puerto Rico, 2019, pp. 514–528. DOI: 10.1145/3359789.3359819.

[18] Zakir Durumeric, David Adrian, Ariana Mirian, Michael Bailey, and J Alex Halderman. "A Search Engine Backed by Internet-Wide Scanning". In: *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*. CCS '15. Denver, CO, USA, 2015, pp. 542–553. DOI: 10.1145/2810103.2813703.

[19] Zakir Durumeric, James Kasten, Michael Bailey, and J Alex Halderman. "Analysis of the HTTPS Certificate Ecosystem". In: *Proceedings of the Conference on Internet Measurement Conference*. IMC '13. Barcelona, Spain, 2013, pp. 291–304. DOI: 10.1145/2504730.2504755.

[20] Zakir Durumeric, Frank Li, James Kasten, Johanna Amann, Jethro Beekman, Mathias Payer, Nicolas Weaver, David Adrian, Vern Paxson, Michael Bailey, and J Alex Halderman. "The Matter of Heartbleed". In: *Proceedings of the Conference on Internet Measurement Conference*. IMC '14. Vancouver, BC, Canada, 2014, pp. 475–488. DOI: 10.1145/2663716.2663755.

[21] Zakir Durumeric, Eric Wustrow, and J Alex Halderman. “ZMap: Fast Internet-wide Scanning and Its Security Applications”. In: *Proceedings of the USENIX Conference on Security*. SEC ’13. Washington, D.C., USA, Aug. 2013, pp. 605–620. ISBN: 978-1-931971-03-4.

[22] Adrienne Porter Felt, Richard Barnes, April King, Chris Palmer, Chris Bentzel, and Parisa Tabriz. “Measuring HTTPS Adoption on the Web”. In: *Proceedings of the USENIX Conference on Security Symposium*. SEC ’17. Vancouver, BC, Canada, Aug. 2017, pp. 1323–1338. ISBN: 978-1-931971-40-9.

[23] Alan O Freier, Philip Karlton, and Paul C Kocher. *The Secure Sockets Layer (SSL) Protocol Version 3.0*. RFC 6101. RFC Editor, Aug. 2011. DOI: 10.17487/RFC6101.

[24] Peipei Fu, Zhen Li, Gang Xiong, Zigang Cao, and Cuicui Kang. “SSL/TLS Security Exploration Through X.509 Certificate’s Life Cycle Measurement”. In: *IEEE Symposium on Computers and Communications (ISCC)*. Natal, Brazil, June 2018, pp. 652–655. DOI: 10.1109/ISCC.2018.8538533.

[25] Slava Galperin, Dr. Carlisle Adams, Michael Myers, Rich Ankney, and Ambarish N Malpani. *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP*. RFC 2560. RFC Editor, June 1999. DOI: 10.17487/RFC2560.

[26] *Guidelines For The Issuance And Management Of Extended Validation Certificates*. Version 1.7.2. CA/Browser Forum, 2020. URL: https://cabforum.org/wp-content/uploads/CA-Browser-Forum-EV-Guidelines-v1.7.2.pdf.

[27] Hwanjo Heo and Seungwon Shin. “Who is Knocking on the Telnet Port: A Large-Scale Empirical Study of Network Scanning”. In: *Proceedings of the Asia Conference on Computer and Communications Security*. ASIACCS ’18. Incheon, Republic of Korea, 2018, pp. 625–636. DOI: 10.1145/3196494.3196537.

[28] Jacob Hoffman-Andrews. *2020.02.29 CAA Rechecking Bug*. Let’s Encrypt Community Support. 2020. URL: https://community.letsencrypt.org/t/2020-02-29-caa-rechecking-bug/114591 (visited on 03/19/2020).

[29] Jacob Hoffman-Andrews. *Let’s Encrypt: CAA Rechecking bug*. Bugzilla. 2020. URL: https://bugzilla.mozilla.org/show_bug.cgi?id=1619047 (visited on 03/19/2020).

[30] 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: *Proceedings of the ACM SIGCOMM Conference on Internet Measurement Conference*. IMC ’11. Berlin, Germany, 2011, pp. 427–444. DOI: 10.1145/2068816.2068856.

[31] Simon Josefsson. *The Base16, Base32, and Base64 Data Encodings*. RFC 4648. RFC Editor, Oct. 2006. DOI: 10.17487/RFC4648.

[32] Simon Josefsson and Sean Leonard. *Textual Encodings of PKIX, PKCS, and CMS Structures*. RFC 7468. RFC Editor, Apr. 2015. DOI: 10.17487/RFC7468.

[33] Stephen Kent. *Privacy Enhancement for Internet Electronic Mail: Part II: Certificate-Based Key Management*. RFC 1422. RFC Editor, Feb. 1993. DOI: 10.17487/RFC1422.

[34] James Larisch, David Choffnes, Dave Levin, Bruce M. Maggs, Alan Mislove, and Christo Wilson. “CRLite: A Scalable System for Pushing All TLS Revocations to All Browsers”. In: *IEEE Symposium on Security and Privacy (SP)*. SP ’17. San Jose, CA, USA, May 2017, pp. 539–556. DOI: 10.1109/SP.2017.17.

[35] John Linn. *Privacy Enhancement for Internet Electronic Mail: Part I: Message Encryption and Authentication Procedures*. RFC 1421. RFC Editor, Feb. 1993. DOI: 10.17487/RFC1421.

[36] Yabing Liu, Will Tome, Liang Zhang, David Choffnes, Dave Levin, Bruce Maggs, Alan Mislove, Aaron Schulman, and Christo Wilson. “An End-to-End Measurement of Certificate Revocation in the Web’s PKI”. In: *Proceedings of the Internet Measurement Conference*. IMC ’15. Tokyo, Japan, 2015, pp. 183–196. DOI: `10.1145/2815675.2815685`.

[37] *Marking HTTP As Non-Secure*. The Chromium Projects. URL: `https://www.chromium.org/Home/chromium-security/marking-http-as-non-secure` (visited on 05/11/2020).

[38] Ariana Mirian, Christopher Thompson, Stefan Savage, Geoffrey M Voelker, and Adrienne Porter Felt. *HTTPS Adoption in the Longtail*. Tech. rep. Google and UC San Diego, 2018. URL: `https://research.google/pubs/pub49037/`.

[39] Austin Murdock, Frank Li, Paul Bramsen, Zakir Durumeric, and Vern Paxson. “Target Generation for Internet-Wide IPv6 Scanning”. In: *Proceedings of the Internet Measurement Conference*. IMC ’17. London, United Kingdom, 2017, pp. 242–253. DOI: `10.1145/3131365.3131405`.

[40] Matus Nemec, Dusan Klinec, Petr Svenda, Peter Sekan, and Vashek Matyas. “Measuring Popularity of Cryptographic Libraries in Internet-Wide Scans”. In: *Proceedings of the Annual Computer Security Applications Conference*. ACSAC ’17. Orlando, FL, USA, 2017, pp. 162–175. DOI: `10.1145/3134600.3134612`.

[41] Henrik Nielsen, Roy T Fielding, and Tim Berners-Lee. *Hypertext Transfer Protocol – HTTP/1.0*. RFC 1945. RFC Editor, May 1996. DOI: `10.17487/RFC1945`.

[42] Gustaf Ouvrier, Michel Laterman, Martin Arlitt, and Niklas Carlsson. “Characterizing the HTTPS Trust Landscape: A Passive View from the Edge”. In: *IEEE Communications Magazine* 55.7 (July 2017), pp. 36–42. DOI: `10.1109/MCOM.2017.1600981`.

[43] *Project Sonar*. Rapid7.com. URL: `https://www.rapid7.com/research/project-sonar/` (visited on 04/24/2020).

[44] Eric Rescorla. *HTTP Over TLS*. RFC 2818. RFC Editor, May 2000. DOI: `10.17487/RFC2818`.

[45] Eric Rescorla. *The Transport Layer Security (TLS) Protocol Version 1.3*. RFC 8446. RFC Editor, Aug. 2018. DOI: `10.17487/RFC8446`.

[46] *Revoking certain certificates on March 4*. Let’s Encrypt Community Support. 2020. URL: `https://community.letsencrypt.org/t/revoking-certain-certificates-on-march-4/114864` (visited on 03/19/2020).

[47] *Scanning All The Things*. Rapid7 Blog. 2013. URL: `https://blog.rapid7.com/2013/09/26/internet-wide-probing-rapid7-sonar/` (visited on 04/04/2020).

[48] Emily Schechter. *A secure web is here to stay*. Google Online Security Blog. 2018. URL: `https://security.googleblog.com/2018/02/a-secure-web-is-here-to-stay.html`.

[49] Trevor Smith, Luke Dickinson, and Kent Seamons. “Let’s Revoke: Scalable Global Certificate Revocation”. In: *Proceedings of the Network and Distributed System Security Symposium*. NDSS ’20. San Diego, CA, USA, 2020. DOI: `10.14722/ndss.2020.24084`.

[50] *SSL Certificates | Rapid7 Open Data*. Rapid7.com. URL: `https://opendata.rapid7.com/sonar.ssl/` (visited on 05/06/2020).

[51] Christian Tiefenau, Emanuel von Zezschwitz, Maximilian Häring, Katharina Krombholz, and Matthew Smith. “A Usability Evaluation of Let’s Encrypt and Certbot: Usable Security Done Right”. In: *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*. CCS ’19. London, United Kingdom, 2019, pp. 1971–1988. DOI: `10.1145/3319535.3363220`.

[52] Emin Topalovic, Brennan Saeta, Lin-Shung Huang, Collin Jackson, and Dan Boneh. "Towards Short-Lived Certificates". In: *IEEE Oakland Web 2.0 Security and Privacy*. W2SP '12. San Francisco, CA, USA, 2012. URL: http://www.ieee-security.org/TC/W2SP/2012/papers/w2sp12-final9.pdf.

[53] Martino Trevisan, Danilo Giordano, Idilio Drago, Maurizio Matteo Munafò, and Marco Mellia. "Five Years at the Edge: Watching Internet From the ISP Network". In: *IEEE/ACM Transactions on Networking* 28.2 (Apr. 2020), pp. 561–574. DOI: 10.1109/TNET.2020.2967588.

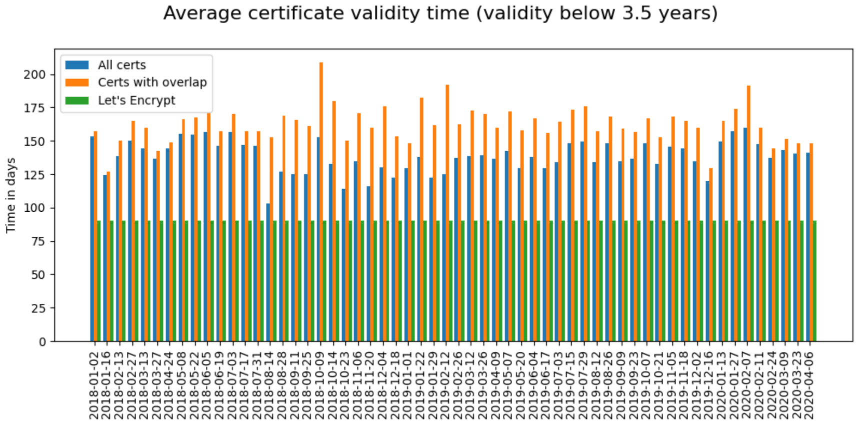
[54] *X.690 : Information technology - ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)*. ITU-T Recommendation X.690. 2015. URL: https://www.itu.int/rec/T-REC-X.690-201508-I.

[55] Liang Zhang, David Choffnes, Dave Levin, Tudor Dumitra\cs, Alan Mislove, Aaron Schulman, and Christo Wilson. "Analysis of SSL Certificate Reissues and Revocations in the Wake of Heartbleed". In: *Proceedings of the Conference on Internet Measurement Conference*. IMC '14. Vancouver, BC, Canada, 2014, pp. 489–502. DOI: 10.1145/2663716.2663758.

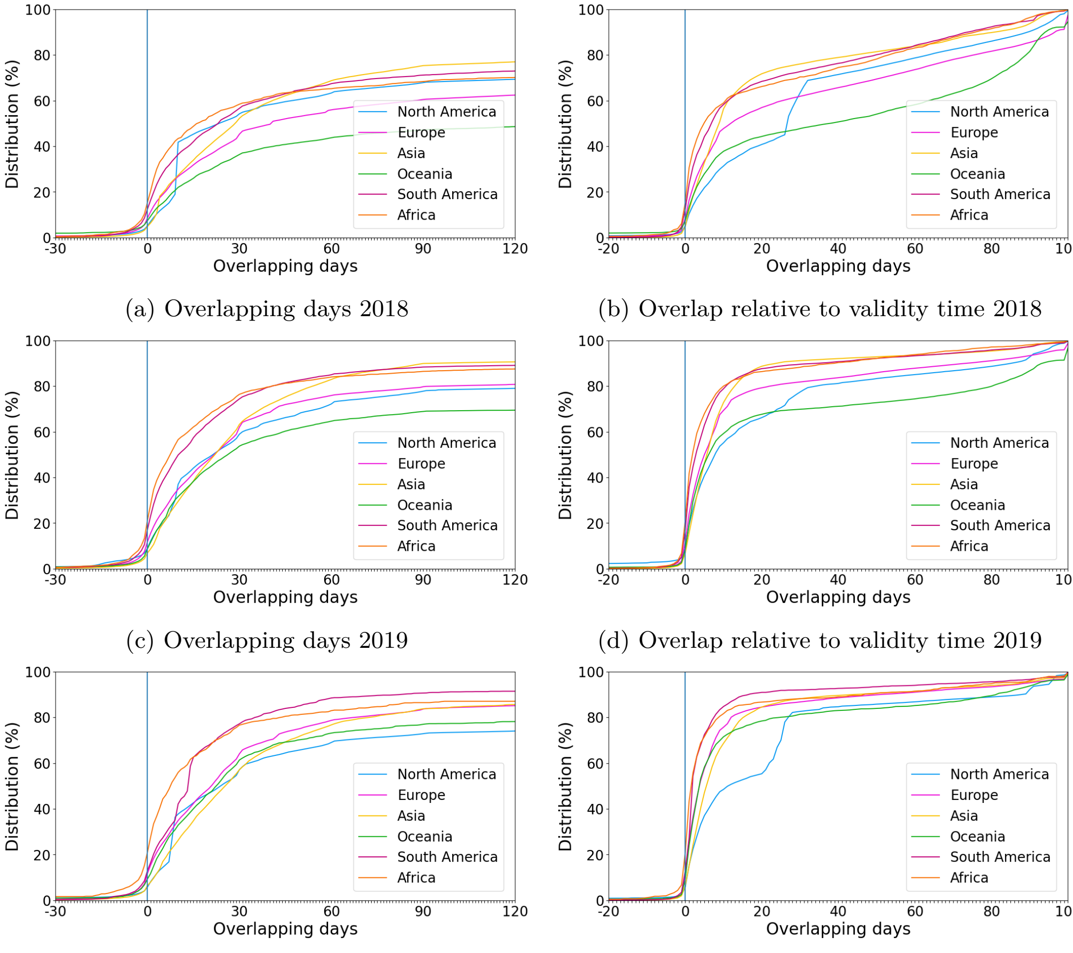
A Appendix

A.1 Additional charts

Trend graphs



Subject Continent



(a) Overlapping days 2018

(b) Overlap relative to validity time 2018

(c) Overlapping days 2019

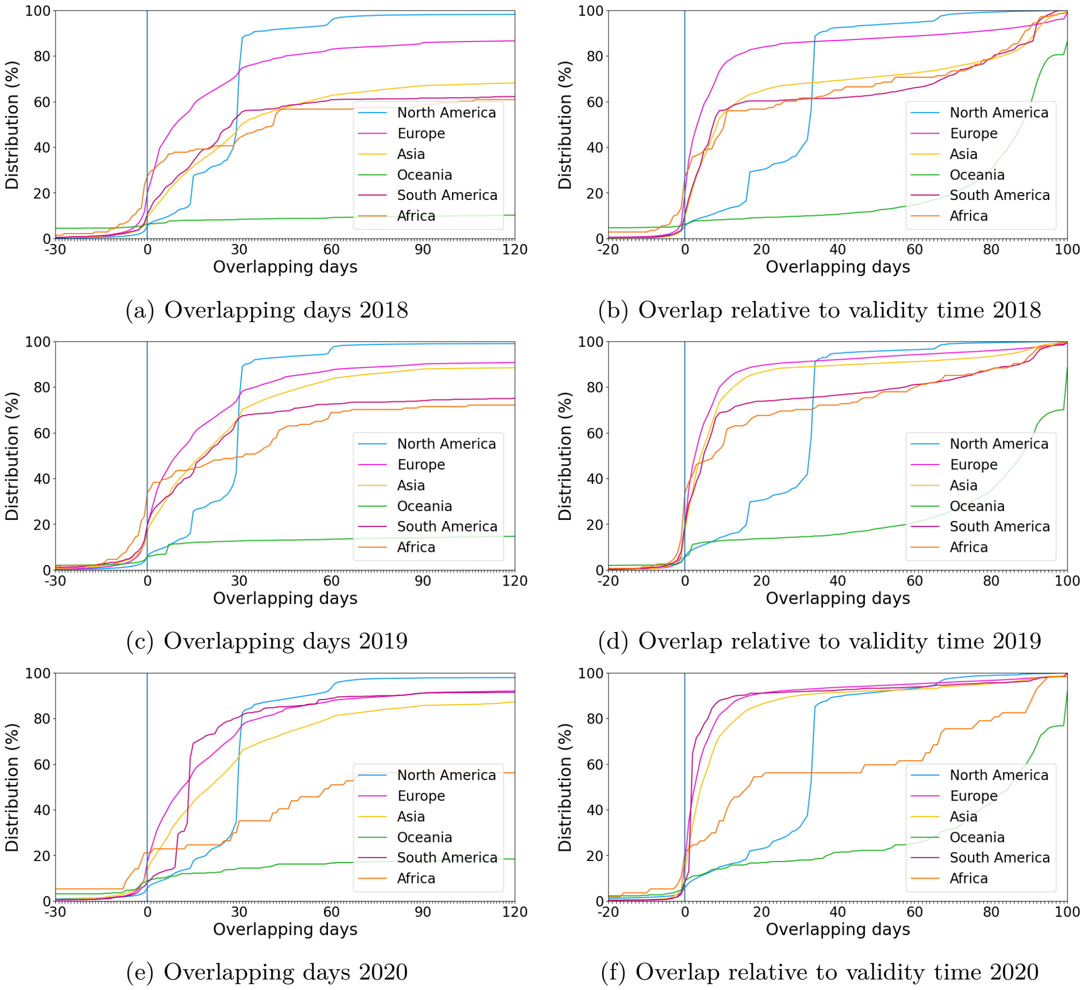
(d) Overlap relative to validity time 2019

(e) Overlapping days 2020

(f) Overlap relative to validity time 2020

Figure A.1: Certificate overlap distribution with regards to subject continent

Issuer Continent



(a) Overlapping days 2018

(b) Overlap relative to validity time 2018

(c) Overlapping days 2019

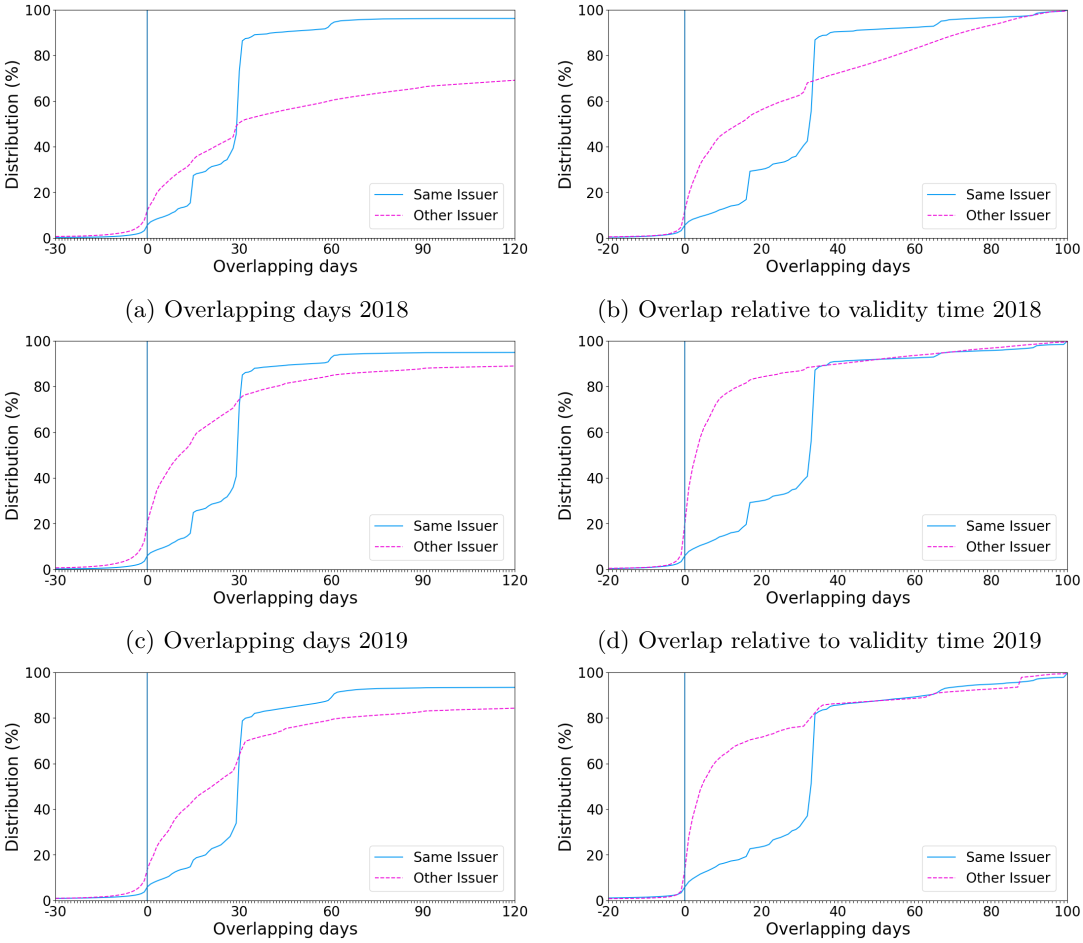
(d) Overlap relative to validity time 2019

(e) Overlapping days 2020

(f) Overlap relative to validity time 2020

Figure A.2: Certificate overlap distribution with regards to issuer continent

Keeping or Changing Issuer



(a) Overlapping days 2018

(b) Overlap relative to validity time 2018

(c) Overlapping days 2019

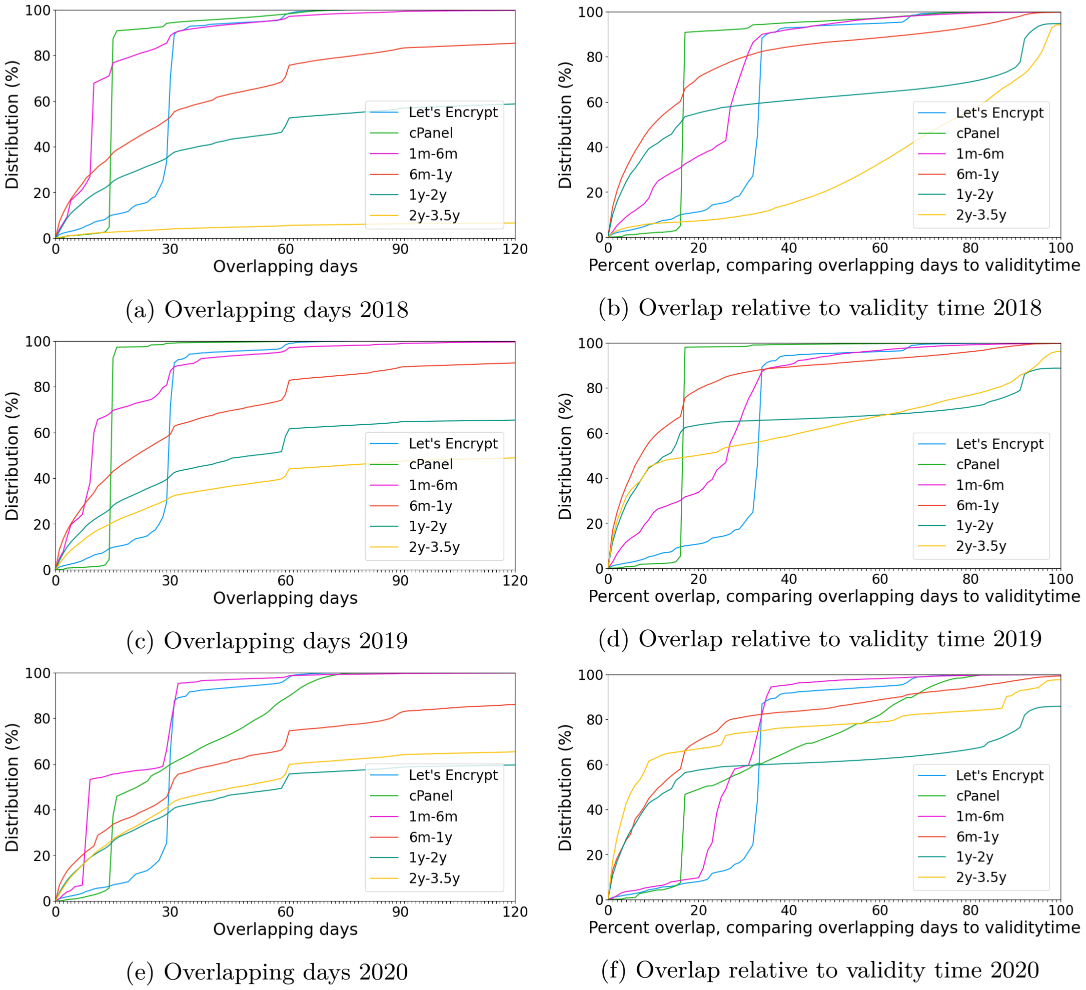
(d) Overlap relative to validity time 2019

(e) Overlapping days 2020

(f) Overlap relative to validity time 2020

Figure A.3: Certificate overlap distribution with regards to loyalty towards issuer

Validity time



(a) Overlapping days 2018

(b) Overlap relative to validity time 2018

(c) Overlapping days 2019

(d) Overlap relative to validity time 2019

(e) Overlapping days 2020

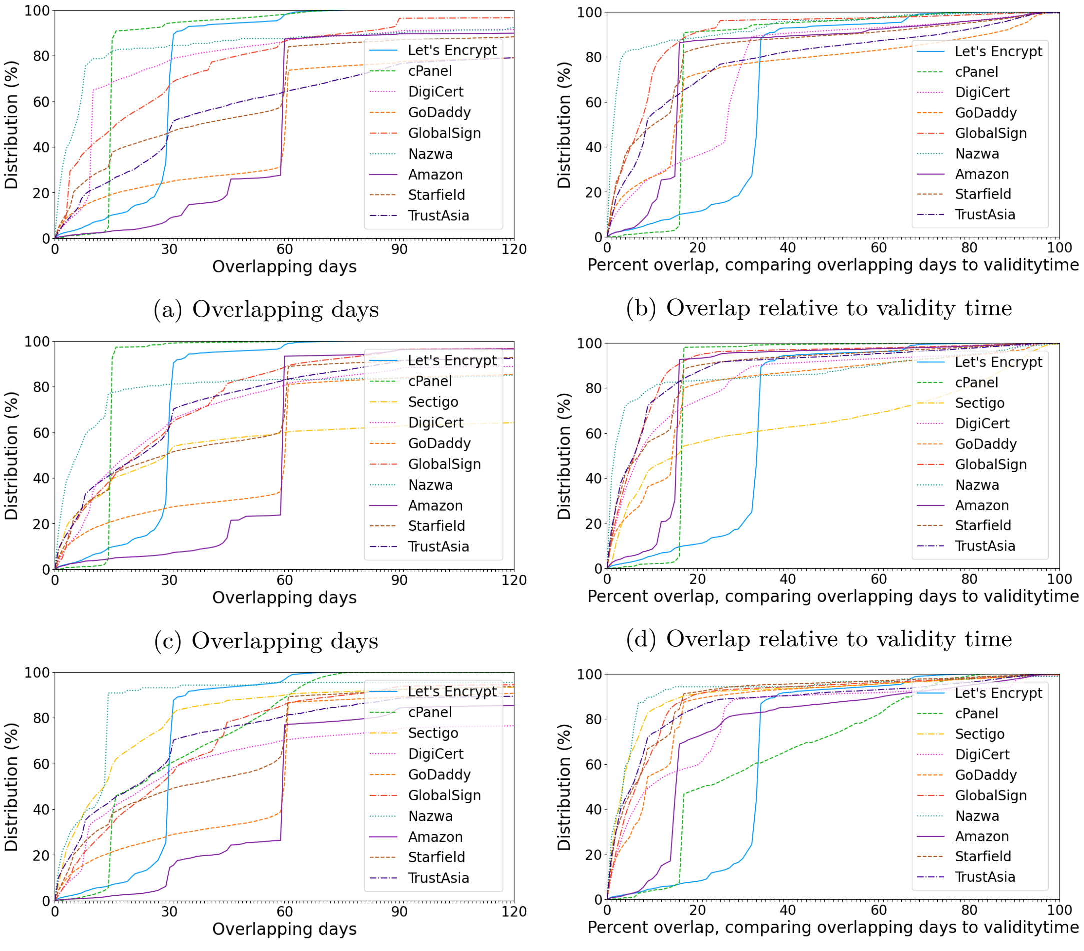
(f) Overlap relative to validity time 2020

Figure A.4: Certificate overlap distribution with regards to validity time

	2018		2019		2020	
Group	**%**	**Count**	**%**	**Count**	**%**	**Count**
Let's Encrypt	73.51%	5 130 708	72.88%	5 785 091	70.78%	1 470 701
1d - 2v	0.16%	11 095	0.05%	4 337	0.04%	875
2v - 1m	0.05%	3 156	0.03%	2 600	0.02%	454
1m - 6m	14.56%	1 015 955	9.87%	783 849	8.84%	183 727
6m - 1y	6.65%	464 452	8.52%	676 678	8.65%	179 739
1y - 2y	4.00%	279 157	7.53%	597 904	8.14%	169 091
2y - 3.5y	1.07%	75 024	1.10%	87 712	3.53%	73 325

Table A.1: Number of certificate, grouped by validity time, over time

Ten Selected Top-Issuers



(a) Overlapping days

(b) Overlap relative to validity time

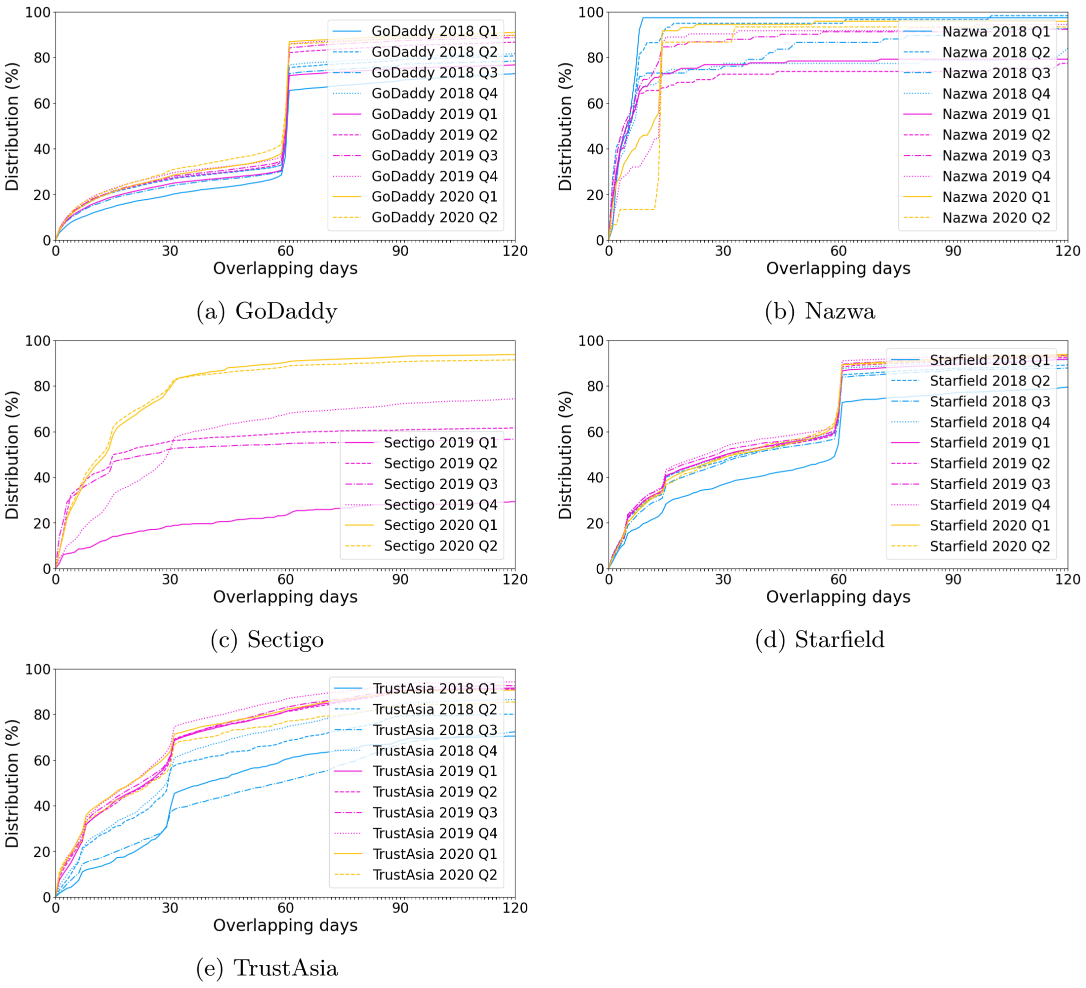
(c) Overlapping days

(d) Overlap relative to validity time

(e) Overlapping days

(f) Overlap relative to validity time

Figure A.5: Certificate overlap distribution for ten selected issuers



(a) GoDaddy

(b) Nazwa

(c) Sectigo

(d) Starfield

(e) TrustAsia

Figure A.6: Certificate overlap distribution of top-performing issuers on a quarterly basis

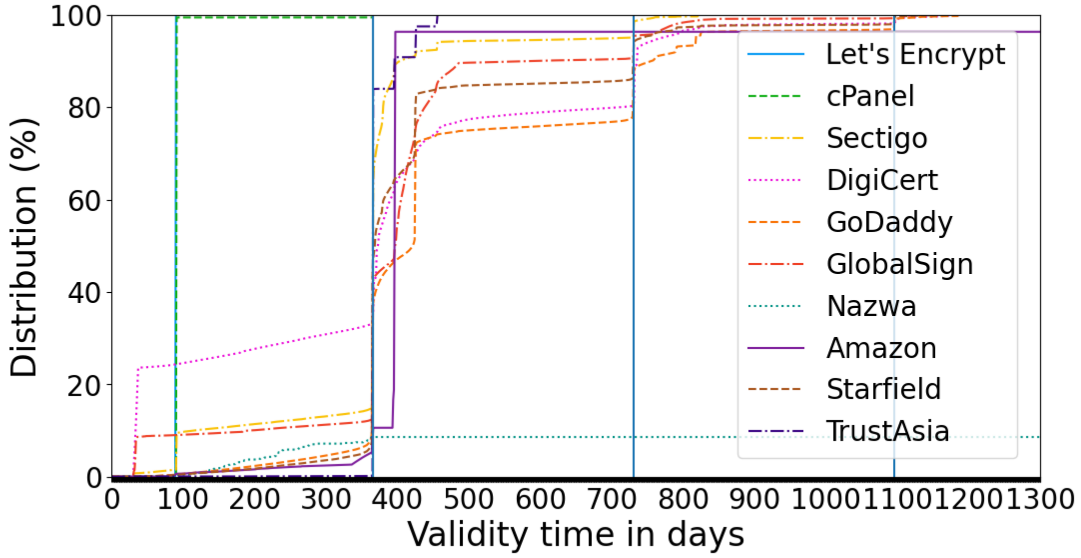


Figure A.7: Distribution of certificate validity time per issuer

A.2 Additional tables

Group	Num. certificates analyzed	Num. replacement relations found
2017	28 119 936	–
2018	34 614 524	11 573 437
2019	76 115 143	18 403 026
2020	65 950 123	6 620 358
Total	204 799 726	36 596 821
Pending	9 647 471	1 557 632

Table A.2: The dataset in numbers

In Table A.2, ”Pending” refers to the certificates that died at the last observed date. These are excluded from the analysis in this thesis as it cannot be determined whether they really died at that date, as that would require a later data point. Pending is therefore certificates dated 2020-04-22 and replacement relations dated 2020-04-06, the last data points in each respective dataset.