Analysing Certificate Transparency logs for Let's Encrypt customer behaviour

Tom Grooters
University of Twente
P.O. Box 217, 7500AE Enschede
The Netherlands
t.grooters@student.utwente.nl

ABSTRACT

TLS certificates are what is keeping the secure web running, they are the cornerstone of secure HTTPS connections. In this paper we will dive into one of the suppliers of TLS certificates, Let's Encrypt. They supply their certificates for free and do this at a 90 day validity period after which users need to renew their certificate(s).
However, how often do users renew their certificates and are they renewing them for extended periods? Not much is known about this as of yet and this research will answer these questions. Using Certificate Transparency (CT) logs, massive collections of all certificates handed out by a certificate authority, we will try to analyze the user behaviour to find this out. In this research, we find that users generally do this (shortly) after a 60 interval, as recommended by Let's Encrypt.

Keywords

Certificate Transparency, TLS certificates, Let's Encrypt, Analysis

1. INTRODUCTION

TLS certificates keep the web running. They are the keystone in offering secure HTTPS connections. These certificates are offered by a Certificate Authority (CA) and need to be trusted. However, we need to know that we can trust the CA. Not every CA has the same standards and is as trustworthy as others. To keep a better view of the actions of a CA and to check on their trustworthiness Certificate Transparency[10] (CT) was introduced.

With Certificate Transparency the provisioned certificates are added to a so-called CT log. Only new certificates can be added to these logs and none can be removed (append-only). The append-only nature is proofed through the usage of a Merkle hash tree[3]. These logs are public and can be viewed and audited by anyone.

One CA that has been very popular and widely used since their introduction to the market is Let's Encrypt[11]. They offer a free service for provisioning these TLS certificates. However, they have imposed some restrictions on their provisioning[5]. Their certificates are valid for only 90 days and are recommended to be renewed every 60 days. However, how well do users keep to these recommendations?

This research will look at how users of Let's Encrypt behave regarding renewals. By answering the research question "*What is the Let's Encrypt customer behaviour around renewing certificates?*" More specifically a look will be taken at the following sub-questions;

- After how many days do users renew their certificates?
- What proportion of issued certificates are being renewed?
- Are the renewed certificates being renewed for longer periods?

It is important for Let's Encrypt that their users do not renew their certificates unnecessarily often. Doing so will result in extra load (and costs) on the CT logs but also on Let's Encrypt themselves. The security benefits are small which makes it important for Let's Encrypt that their users follow the guidelines.

To the best of our knowledge, this is the first large scale analysis of Let's Encrypt user behaviour based on Certificate Transparency logs.

33rd Twente Student Conference on IT July. 3rd, 2020, Enschede, The Netherlands.

2. BACKGROUND

2.1 Certificates

TLS certificates are issued by certificate authorities. However, there is a lot of trust in these CA's. Since they can issue a certificate in the name of any website. To make sure that it is known which CA's can be trusted there is a chain of trust. In simple terms, this means that there are a few known trusted entities that decide whether or not the lower ranking entities can be trusted.

2.2 X.509

TLS certificates use the X.509 standard[6] For their structure. A basic display of the layout can be found in Table 1. Some parts of the structure are version-specific. The current (widely used) version is version 3. However, the first version supplies all required data and later versions add onto this. This data includes the validity period and the subject. The subject tells us which site the certificate supports (through a Common Name field) and the validity period allows us to get an indication of when a certificate was requested (through a not_before field).

Version	
Serial Number	
Signature	
Issuer	
Validity	Valid from
	Valid until
Subject	
Subject Public Key Info	
Issuer Unique ID (version 2+)	
Subject Unique ID (version 2+)	
Extensions (version 3+)	

Table 1: Basic subset of the X.509 structure

2.3 Certificate Transparency

However, how do you detect a CA that has been compromised or gone rogue? For this the Certificate Transparency[10] (CT) logs have been set up. These are public logs that can be viewed and audited by anyone. They are also append-only such that their integrity remains intact. To further help integrity, the use of Merkle hash trees guarantees that only new certificates can be added and that removing previously published data is noticed.

Log availability is also an important factor. For this reason, it is recommended to include proofs from at least three CT logs[2]. Some major players in the field that host their own logs are;

- Cloudflare
- DigiCert
- Google
- Let's Encrypt

2.4 Let's Encrypt

One such Certificate Authority which also operates a Certificate Transparency log is Let's Encrypt. They are one of the largest providers of certificates and do this free of charge. This has been a suspected major factor in their growth and popularity. Especially among (but not limited to) "less-popular domains and in countries with traditionally lower Internet penetration"[11].

Let's Encrypt also have an automated script called Certbot. This program can automatically be run at a set interval and will automatically check which certificates need to be renewed and will perform this action for the user. Next to Certbot they also provide the option of renewal reminders when you provide your email. These will be sent 20 days, 10 days and 1 day[1] before a certificate that has not yet been renewed will expire.

3. RELATED WORK

To the best of our knowledge, no previous works are looking into Let's Encrypt user behaviour using Certificate Transparency logs. However, other works are looking in a broader aspect at Let's Encrypt itself and also into a more general case of Certificate Transparency[9].

Previous works[11] show that Let's Encrypt has a high adoption rate. With over one billion certificates supplied[4] over nearly 192 million domains as of February 2020. This means that a lot of certificates will be available and should provide sufficient data for the research to be conducted.

Certificate Transparency can reveal a lot of data, both good[7] and bad[12]. It can help in the battle against phishing, by finding the domains scammers use in their phishing practices, But on the other hand, as Roberts finds, it can also offer up a lot of unintended data through the domains used for certificates. It can leak all sorts of data like phone numbers, names and even company relations. Another research finds that extracting this data poses little overhead and should be able to be done quickly and efficiently[8].

4. METHODOLOGY

To answer these questions the Certificate Transparency logs of Let's Encrypt will be analysed for this information. The analysis will be done with the use of Apache Spark on a Hadoop cluster.

4.1 Data sets

To get to the information about Lets Encrypt usage behaviour the Certificate Transparency (CT) logs provided by Let's Encrypt themselves are going to be used. This data set contains information on 900 million certificates starting in early 2019. By running PySpark on a Hadoop cluster this information can be analysed efficiently. For even more efficient analysis, some data was pre-formatted to be stored in a parquet format with some data pre-extracted for more efficient analysis. This includes the Common Names, not_before timestamp and the issuer, which are all stored in the certificates.

4.2 Limitations

Due to the way the data is provided, there are some limitations this research is bound by. The main hurdle is the fact that the exact issuance timestamp is not provided. To get around this part usage will be made of the not_before flag. Since the certificates issued by Let's Encrypt are valid shortly after requesting such a certificate, it can be reasonably assumed that the not_before timestamp is a good indicator of when a certificate was requested. As such this timestamp will be used as the issuance/request timestamp in terms of intervals.

Besides the lack of the issuance time, it is also not certain which certificates follow each other. To get around this limitation usage will be made of the Common Names (CN) in the certificate. These show for which domains the certificates are valid. By grouping certificates together based on the CNs the hope is to reliably get around this limitation and still get a good indication of the renewal process.

4.3 Intervals

As described in Section 4.2 there is a slight limitation on the accuracy of the timestamps. However, since the timescale that we will be working on is days this small uncertainty should not have a meaningful impact on the results. To get the interval between certificates the plan is to group the certificates based on the domains it covers. Although it can occur that a user requests the certificates covering the same (sub)domains, the expectation is that these numbers are negligible. After this grouping, the certificates will be sorted to make sure they are in the correct order after which the interval between the not_before timestamps will be processed.

4.4 Certificate renewal proportion

To get an indication of the proportion of certificates that are being renewed the plan is to look at the number of certificates that are being issued covering the same set of domains. Over each set of certificates belonging to the same grouping, the length will be taken as an indication to what extend the certificates are being renewed. If this length is one it means that only a single certificate was

Duplicates	Occurrences
30+	437
20-29	2466
15-19	3175
14	1546
13	804
12	5896
11	1775
10	59821
9	1770
8	62436
7	2136
6	210022
5	3764
4	2399232
3	38455
2	437915934
1	4614495

Table 2: The number of duplicates. grouped by number of occurrences

requested but if it is greater than one it means multiple certificates have been requested for this same set of domains.

4.5 Certificate renewal duration

After getting the groupings of certificates together we can further analyse this information to extract for what duration certificates are being renewed.

The main focus will lie on the fact whether there will be a clear indication that this is a single year. This is because the standard duration for registering a domain name is one year. As such there is the expectation that there is a noticeable number of domain sets of which the coverage will span around a year, accounting for certificates being renewed shortly before the domain expires.

5. RESULTS

The code used to get these results together with the raw tables can be found at:
`https://github.com/TomGrooters/ResearchResults`

5.1 Duplicate certificates

One problem that quickly came to light is the fact that quite a lot of certificates seem to be issued multiple times. In this case, multiple certificates cover the exact same Common Names(CN), not_before and not_after timestamps while having a unique fingerprint. Additionally, quite a few certificates are requested shortly after each other. Further research must show the reasoning behind this but a possible hypothesis for the same dated certificates is the fact that race conditions could occur in running the certificate requesting. This resulting in multiple servers being contacted (or multiple instances contacting the same server) resulting in multiple certificates getting registered at the same time. A theory for the case of multiple certificates being issued shortly after each other (in the hours range up to a week) is that inexperienced users may request a certificate multiple times. Either due to misconfiguration, misunderstanding or having a wrong setup.

One hint towards the actual cause is that it seems that even numbers occur most often. And the overwhelming majority occurs twice, 437.915.934(*2) out of a total of 934.348.836 certificates issued by Let's Encrypt. A more complete overview of the numbers can be found in Table 2.

5.2 Filtering duplicate certificates

To work around this and still be able to extract useful data a filter was added, See Figure 1 for the code. This filter filters out all certificates that fall within a 30-day interval of the last accepted one. This filter works on the not_before timestamps which are the best approximation to the issuance time available. The function receives a list of all timestamps as input. The function first sorts this list to get them in the correct order, after which the first timestamp is added to the output list and the most recent accepted certificate is stored in a variable. After this initial setup, a for loop is entered which loops through all the other timestamps. When it finds a timestamp that surpasses the last accepted one + the interval the timestamp gets added to the output list and the most recent timestamp gets set to the current found one.

```
# 1 month in seconds
interval = 60*60*24*30

def filter_duration(timestamps):
    timestamps.sort()
    result = [timestamps[0]]
    last = timestamps[0]
    for o in timestamps[1:]:
        if last+interval < o:
            result.append(o)
            last = o
    return result
```

Figure 1: Python code to filter on interval

5.3 Missing certificates

There are also strong suspicions towards missing certificates. Looking at the renewal intervals there are peaks in the intervals around multiples of 60 days (120, 180, 240, 300). This is clearly shown in the expanded view of the certificates in Figure 2d This strongly indicates towards these certificates being renewed in 60 days intervals but are missing certificates in between (the lack of a peak around 270 suggest that these are not at a 30 or 90 day interval). Whether these are missing because the dataset is incomplete or if these are missing from the CT logs has to be determined.

5.4 Renewal interval

For investigating the interval between renewals there are multiple ways to count this data. As a result, multiple methods have been applied. This data has been put in graphs in Figure 2, these tables have been cut off at 30 (by the aforementioned filter) and 120 days to keep a better overview.

- Getting the average per group of domains.
- Getting the median per group of domains.
- Getting all intervals.

Getting all intervals looks at each consecutive certificate regardless of the number per domain. This means that in the case of 10 certificates for one set and 4 for another the case with 10 certificates has a stronger impact. However, this will give a more complete overview, at a cost of fairness.

Looking into this data it shows that the overwhelming majority of certificates are renewed (shortly) after the 60 days recommended period. This also coincides with the standard time the Let's Encrypt Certbot renews the certificate. There are also noticeable peaks at the 70 day mark (which

Certificates	Occurrences
16	1415
15	3645
14	8156
13	33508
12	61555
11	100242
10	152771
9	452949
8	2719206
7	11336882
6	19167689
5	11265436
4	11739511
3	13268880
2	16922799
1	47432163

Table 3: The number of certificates per grouped domains (filtered to a minimum of 30 days apart)

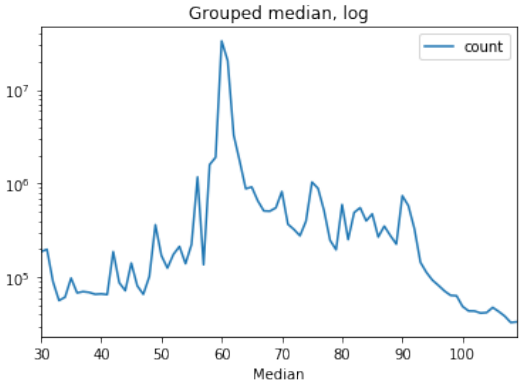
is 20 days before expiration, which is when Let's Encrypt sends a renewal reminder. As well as around the 90 day mark when the certificate actually expires.

5.5 Renewal proportion

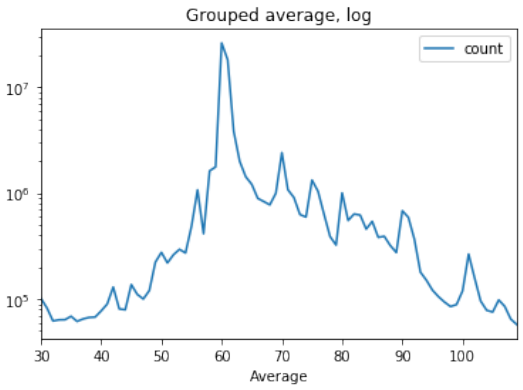
To know the proportion of certificates that are being renewed a look is taken at the number of certificates for each grouping of Common Names. By taking this count a good indication is gotten of whether or not (and how often) a certificate has been renewed. In this case, a count of one (which means that there is a single certificate) indicates that the certificate was never renewed. Analyses show that of the 134.666.807 sets of CNs being covered 47.432.163 sets only have one certificate. This comes down to 35.22% of the certificates being renewed. See Table 3 for the full numbers. As always this has to be taken with a grain of salt since it is not known whether all issued certificates are being shared by Let's Encrypt.

5.6 Renewal duration

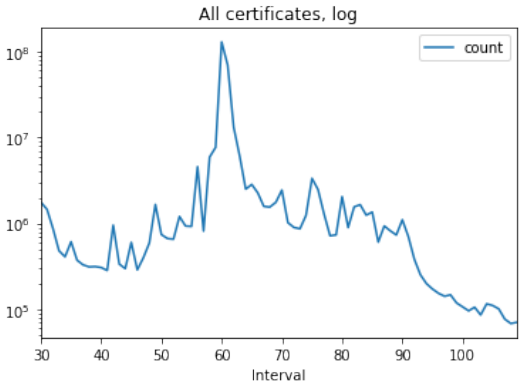
The duration of renewals is a tricky thing to discover. It turns out that the coverage of the data is smaller than expected and the impact is also underestimated. As a result, the goal of detecting the impact of the 1 year domain registration period will be severely hampered. However, an effort can still be made on shorter instances. Therefore a look will be taken at short duration certificates (far below the 1 year domain period) and the focus will lie on certificates that have been renewed for a short duration. The first thing that stands out is that besides the large group of certificates that are one-offs there are also large groups of CN sets that have two or three certificates. Looking deeper into these statistics of when these occur (to account for previous and future certificate renewals outside the data scope) shows that most of these domains indeed fall in the middle of the data. This shows that it is not uncommon to have certificates active for a short period, besides only requesting one certificate. This shows that many short term projects also make use of these Let's Encrypt certificates. Another noteworthy fact is that a large set of certificates are renewed far more than they should. Up to almost monthly. However, this can also be a result of our limitations. It is not unheard of that users have the same domains on multiple servers and requesting new certificates for each. This is something future works might be able to take a deeper look at.



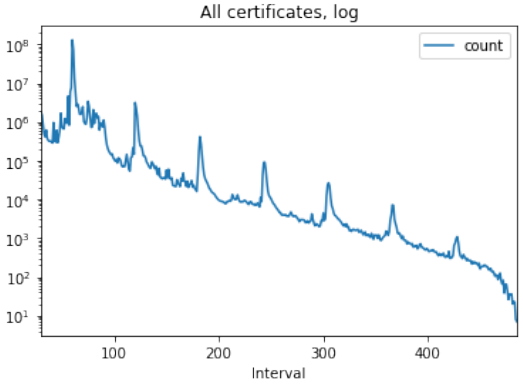
(a) Median of grouped CNs



(b) Average of grouped CNs



(c) Renewal interval between certificates, logarithmic scales



(d) Renewal interval between all certificates

Figure 2: Overview of certificate renewal time in days
X axis: the number of days
Y axis: The number of certificates, logarithmic scale

6. CONCLUSION

To answer the research question *What is the Let's Encrypt customer behaviour around renewing certificates?* we first look at the sub-questions.

To answer the first research question "*After how many days do users renew their certificates?*" the answer seems to be overwhelmingly after around the 60-day recommended threshold. Allowing for some margins in the automated service being run periodically the data show that by a very large margin most certificates are renewed after 60 days or shortly thereafter. There is also a noticeable bump in the 20-day range when Let's Encrypt sends a reminder email and at the 90-day mark when certificates expire. Certificate renewal seems to be happening around the 60-day mark in the overwhelming majority of cases. Based on the interval data between certificate renewals. Although not all data lines up perfectly a likely explanation on this occurrence is the fact that a program must be run and this can result in these few days of delay.

Answering the second question of "*What proportion of issued certificates are being renewed?*" the answer is a little more complicated. Multiple statistics can be used in this case but the main one is the number of certificates covering the same Common Names. This results in 47.432.163 of the total 134.666.807 sets of CNs being covered. This means that in 35.22% of the cases the certificate are not being renewed.

The third question at hand "*Are the renewed certificates being renewed for longer periods?*" is the most complicated. As it turns out the date range being covered is shorter than expected. This resulting in the fact that no satisfactory conclusion could be had fro the main goal of this question (finding out whether certificates are being renewed for 1 year, the shortest period a domain can be registered for, or not). However, the numbers do hint towards a lot of certificates only being used for a short period.

In conclusion, we find that the majority of users keep themselves to the recommended practices, of renewing certificates after 60 days. We also find that in over a third of cases (35.22%) the certificates are only being used for a single period. Sadly there is no clear indication of how long certificates are being renewed for, whether this is a single year or multiple years in a row.

As a further result, we find that analysing these logs is not as straightforward as expected. It turns out that there is a lot of redundant data we must sieve through to get to the relevant parts. As a byproduct, this introduces some uncertainty in the results although the found results still give a good idea in the right direction.

Further research

As the first, to the best of our knowledge, research into Let's Encrypt using Certificate Transparency logs. We found that there are a number of unexpected results that can be picked up by future researches. Since a lot of interesting findings have come up during this research there remain a fair share of unanswered questions but also questions that could benefit from further research.
First off the duplicate certificates, which form a concern for the integrity of Certificate Transparency logs. This data might be in error and overwrite the actual data that is supposed to be here.
Secondly is the very short renewal rates for some Common Name groupings. Ranging from hours after each other to larger time spans. It is not certain that these are actually from the same end-user, but this is something further investigation must clear up.
A third option is to look into the greater intervals between certificates at/around multiples of 60 days. This might indicate certificates missing from the transparency log. Also posing a concern to the integrity of said logs.

7. ACKNOWLEDGEMENTS

I would like to thank Dr Jonker for his support and time in supervising and help during this research and granting access to the OpenIntel systems to run the analysis.

8. REFERENCES

[1] Expiration Emails - Let's Encrypt - Free SSL/TLS Certificates. Retrieved from https://letsencrypt.org/docs/expiration-emails/.

[2] FAQ - Certificate Transparency. Retrieved from https://www.certificate-transparency.org/faq.

[3] How Log Proofs Work - Certificate Transparency. Retrieved from https://www.certificate-transparency.org/log-proofs-work.

[4] Let's Encrypt Has Issued a Billion Certificates - Let's Encrypt - Free SSL/TLS Certificates. Retrieved from https://letsencrypt.org/2020/02/27/one-billion-certs.html.

[5] Rate Limits - Let's Encrypt - Free SSL/TLS Certificates. Retrieved from https://letsencrypt.org/docs/rate-limits/.

[6] RFC 5280 - Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. Retrieved from https://tools.ietf.org/html/rfc5280.

[7] E. Fasllija, H. F. Enişer, and B. Prünster. Phish-Hook: Detecting Phishing Certificates Using Certificate Transparency Logs. In S. Chen, K.-K. R. Choo, X. Fu, W. Lou, and A. Mohaisen, editors, *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, volume 305 LNICST, pages 320–334, Cham, 2019. Springer International Publishing.

[8] D. Kales, O. Omolola, and S. Ramacher. Revisiting user privacy for certificate transparency. In *Proceedings - 4th IEEE European Symposium on Security and Privacy, EURO S and P 2019*, pages 432–447. Institute of Electrical and Electronics Engineers Inc., 2019.

[9] N. Korzhitskii and N. Carlsson. Characterizing the Root Landscape of Certificate Transparency Logs. Technical report.

[10] B. Laurie. Certificate transparency. *Communications of the ACM*, 57(10):40–46, 9 2014.

[11] A. Manousis, R. Ragsdale, B. Draffin, A. Agrawal, and V. Sekar. Shedding Light on the Adoption of Let's Encrypt. 2016.

[12] R. Roberts and D. Levin. When certificate transparency is too transparent: Analyzing information leakage in HTTPS domain names. In *Proceedings of the ACM Conference on Computer and Communications Security*, pages 87–92, New York, New York, USA, 2019. Association for Computing Machinery.