Analyzing Certificate Transparency (CT) logs

Andreas Zeijlon
Email: andze132@student.liu.se
Supervisor: Niklas Carlsson, niklas.carlsson@liu.se
Project Report for Information Security Course
Linköpings universitetet, Sweden

Abstract

This report describes the work that has been done on exploring the Certificate Transparency landscape with focus on the coronavirus outbreak and how it has affected the submission of domain names. The problem to be addressed was to find if there are any trends or spikes regarding branches that might have been affected by the coronavirus outbreak. Work was also put in to find whether phishing domains would, to a greater degree, mimic authentic websites related to the coronavirus to attract more traffic. This was done by analyzing 90+ million certificates that were submitted during the spring of 2020. The results found that the coronavirus has indeed affected the registration of domains with certain keywords however, it could not be shown that phishing domains mimic authentic websites that has seen a rise in popularity because of covid-19 more than before the outbreak.

1. Introduction

The spring of 2020 will go down in the history book as the time the coronavirus outbreak occurred. The coronavirus has affected millions of people around the world. It's not unthinkable to imagine that the crisis has made the traffic of relevant websites increase. Websites like who.int and worldometers.info should logically see an increase in users which naturally also should catch the attention of phishers.

Phishing and social engineering remain the biggest threats in cybersecurity. 62% of businesses experienced phishing and social engineering attacks in 2018 [1]. 32–33% of all security breaches included phishing or social engineering. It is perhaps the most effective way to compromise a system since humans are often the weakest link and easy to manipulate.

To be able to trust that the website you are visiting is the one you think it is and not a phishing site, services use digital certificates to prove their authenticity. Unfortunately, certificates can easily be forged or mistakenly issued to a malicious website. The answer to this problem was Certificate Transparency (CT). CT, developed by Google, is an open-source framework for monitoring digital certificates. It makes certificates accessible to the public for audit to mitigate the risks of certificates being mistakenly or maliciously issued. CT changed and reinvented the playing field for phishing detection tools as it was now possible to get information like domain names from all submitted certificates. It also introduced a new unique way to characterize and analyze the registration of domain names on the internet over time, to find both trends and trend breaks related to, for example the coronavirus.

The purpose of this report was to find any such trends or trend breaks related to the coronavirus and identify a subset of phishing domains taking advantage of the increased user traffic of certain websites related to the coronavirus.

Questions at issue:

- Is there any correlation between the coronavirus chain of incidents and the registration of domain names with certain keywords?
- Is there a trend of phishing domains mimicking authentic websites related to covid-19?

2. Background

In this section key areas needed to understand the rest of the report will be introduced.

2.1 SSL/TSL

SSL (Secure Socket Layer) and TLS (Transport Layer Security) are security communication protocols used for sending encrypted data between two parties [2]. The latter is a further development of SSL version 3 and has since replaced the former as the standard for sending encrypted data. TLS, and SSL, sits between the application layer and transport layer of the TCP/IP-model and can be used by a number of protocols in the application layer like HTTPS, FTPS, IMAP, POP3 and SMTP.

SSL/TLS uses public key cryptography, or asymmetric cryptography which is a system that uses pairs of keys, a public key and a private key [3]. The public key of the recipient is used to encrypt the data being sent and therefore does not need to be kept secure, while the private key of the recipient is used to decrypt that same data and should only be known to the owner/recipient.

One benefit of asymmetric cryptography is that it also allows the sender to digitally sign a message by appending a version encrypted with the its private key. The recipient can then verify the sender by decrypting the encrypted message with the sender's public key and checking that it matches the original message. This verification proves that the sender had access to the private key and therefore is likely to be the subject that the public key is associated with but, how can we trust that the one we are communicating with is the actual owner of the public key? It could be someone else pretending to be an authentic service.

Therefore, to establish the identity of a server or client, SSL and TLS use digital certificates that can prove that an entity is what it claims to be.

2.2 Public key certificates

A public key certificate, or digital certificate, is an electronic document that certifies the ownership of a public key [4]. A digital certificate contains information about the public key, information about the owner and a digital signature signed by a third-party, called *Issuer*, that has verified the contents of the certificate. If the signature is valid and the issuer is trusted, then it is safe to use that key to communicate securely with the owner of the certificate.

In TLS a service is required to present a valid certificate to a connecting client as part of the initial connection setup. A certificate can be self-signed by the service however, when this is the case clients will generally be unable to verify the certificate and reject the connection. Instead the certificate should be signed by a third-party issuer. Typically, the issuer is a Certificate Authority (CA), which is a company that charges customers to issue certificates for them.

This system worked fine until it became evident that CA's had, by being compromised or by mistake, issued certificates for malicious websites. In these cases, clients trust the websites because they trust the CA that signed their certificates. Therefore, giving the users the impression that the website they are visiting is authentic and their connection is secure.

An example of this is back in 2011, when a dutch CA called DigiNotar was compromised [5]. The hackers were able to use DigiNotar's system to issue fake SSL certificates to mimic numerous sites in Iran, which enabled the hackers to spy on unsuspecting users of the fake websites.

This led to the development of a system where Certificate Authorities could be reviewed by the public and consequently mitigate the damage done by fake or mistakenly issued certificates. That system is called Certificate Transparency.

2.3 Certificate transparency

Certificate transparency (CT), developed by Google, aims to reduce the risk of fake certificates causing harm by making all issued certificates public and open for scrutiny [6]. CT makes it impossible for a CA to issue a certificate for a domain without the domain owner knowing about it. It lets domain owners and CAs find out whether certificates have been mistakenly or maliciously issued. It protects users from being duped by certificates that were mistakenly or maliciously issued.

When a certificate is issued, for example by a CA, the issuer also submits the certificate to a CT log. A CT log is an append-only record of issued SSL certificates that is available to the public for review. The contents of a CT log cannot be deleted, altered or retrospectively be inserted. There are numerous CT logs of various sizes and characteristics. Some of Googles own CT logs are operated by web crawlers that find certificates via their search engine and appends them automatically to the log while other CT logs are for manual submission only.

CT does unfortunately not bring only benefits but also vulnerabilities, which opens up to a new range of attacks. By making certificates public, CT logs can be used to find a lot of information about a server or network. For example, identify hidden internal subdomains of a given domain or find newly deployed servers which might not yet have implemented sufficient security.

An example of this was given by a hacker named Hanno Böck at Defcon, who showed that it was possible to take over Wordpress installations and install a backdoor plugin [7]. This was done by monitoring CT logs and identifying freshly deployed Wordpress installers. The attacker could then take over the installation of the application and install a plugin that contained a PHP shell and then revert the installation by deleting the configuration file created by the installer, thus erasing all traces of the plugin and the attack. The owner of the web application wouldn't notice anything as the installer would be available again after the attack. Now the attacker had a backdoor into the application which could be used to execute code.

3. Methodology

The method of work mainly consisted of two parts; literature study and research. In the literature study phase information regarding the subject *Certificate Transparency* was gathered by scouring the internet. Mainly Google search engine was used. The goal of this phase was to build an inventory of interesting things that certificate transparency logs can be used for and figure out a what research can be done with a reasonable scope given the limited time frame. Every source found that was of interest was added to a table for later reference. The main problem that needed to be solved was finding a

feasible and sustainable solution to downloading, storing and analyzing a very large dataset of certificates.

In the research phase a large dataset of certificates, approximately 280+ million, were downloaded and analyzed with regards to the questions at issue and the result was recorded and plotted.

The result of that analysis will be presented in this report. This report was written as the project was ongoing and altered many times.

Tools and materials used in the research was:

- Fuzzyball.js library for levenshtein function
- Axeman
- Google BigQuery
- Google Compute Engine
- Matlibplot python

To analyze certificates from CT logs they first had to be downloaded. Every CT log provides an API to download certificates however, they only allow downloading small blocks of certificates at a time. On top of that the data structure for each result was a fairly opaque binary stream, which had to be parsed. Fortunately, there was an open-source tool called Axeman that could automate the whole process of downloading and parsing all certificates of a given CT log. Axeman retrieves, parses and stores certificates into csv-files automatically using concurrency and multi-processing. To speed up the process of collecting the certificates, Google Compute Engine was used. A virtual machine with 30 GB ram and 16 vCPUs was used to install Axeman and retrieve 280+ million certificates, ~495 GB of data, in approximately 11 hours. The certificates were downloaded 2020-04-10. The CT log chosen was Cloudflare Nimbus 2020 since the data interesting to this project would be from the year 2020. Axeman stored the resulting certificates into csv-files with the following headers:

- *url* (The url of the CT log)
- *cert_index* (The index number of the certificate)
- *chain_hash* (The hash of the certificate chain)
- *cert_der* (A DER sequence of the certificate)
- *all_dns_names* (A string of all DNS names white-space separated)
- *not_before* (An Epoch timestamp the certificate is valid after)
- *not_after* (An Epoch timestamp the certificate is valid before)

The csv-files was then transferred from the virtual machine into a Google storage bucket. To be able to query the large amount of data, Google BigQuery was found to be the best option in terms of speed and stability. BigQuery could import csv-files into a table automatically very fast. The dataset consisted of 280+ million rows, each row corresponding to a certificate, with six columns corresponding the csv-file headers. To bring down the cost of storage and reducing time for queries, certificates that wasn't valid before January 1, 2020 was deleted from the dataset. The resulting dataset consisted of 92 229 991 certificates all valid in 2020.

There were two methods used for analyzing the dataset. Substring matching and string similarity checking using the Levenshtein-distance algorithm.

When doing substring matching every entry of the column *all_dns_names* was compared to a regular expression and if there was a match the whole row would be written into a csv-file. The SQL-query would look like this:

```
'select * FROM `[PROJECT-ID].[DATASET-NAME].
[TABLE-NAME]` where REGEXP_CONTAINS(
all_dns_names, r".*keyword.*")'
```

Then the python library matlibplot was used to plot the result with the *not_before* column serving as x-axis and the number of occurences serving as the y-axis. When deciding what date the certificates were issued, the *not_before* value was used as it is often the case that the *not_before* date is the same date the certificate was submitted however, it is not a certainty.

When performing the Levenshtein-algorithm, BigQuery UDF was utilized. UDF, user-defined functions, makes it possible to perform JavaScript code on rows in the dataset. The Levenshtein algorithm was imported from the open-source library fuzzyball.js and then implemented with JavaScript. Since the column *all_dns_names* could consist of several whitespace-separated domain names they first needed to be split up before passing them to the Levenshtein function. Below is how the UDF was implemented in BigQuery.

```
CREATE OR REPLACE FUNCTION
[PROJECT_ID].[DATASET-NAME].levenshtein(a
string, b string)
RETURNS INT64
LANGUAGE js AS """
var dns_names = a.split(' ');
var shortest_distance = 999;
for (var i = 0; i < dns_names.length; i++){
   dns = dns_names[i];
   var dist = fuzzball.distance(dns, b);
   if(dist<shortest_distance){
      shortest_distance = dist;
   }
}
return shortest_distance;
"""
OPTIONS (library="gs://[BUCKET-
NAME]/fuzzball.umd.min.js");
```

The data collection and processing were first attempted locally using a MySQL database however,

because the dataset was larger than the available RAM on the hardware, querying was extremely slow.

Hardware limitations made it infeasible to store and analyze the certificates locally. It simply took too long to process that many certificates. Therefore, Google Cloud was chosen for its processing and storage capabilities. Google Cloud offers 300$ as free trial which was enough for this project.

For the string similarity check, the Levenshtein algorithm was chosen as it seemed to be the easiest to implement while still giving a good enough result when searching for similar strings.

4. Result

In this section the result of the research is presented.

4.1 Substring matching

The result of the substring matching using regular expressions showed both promising and mixed results.

In the first test case the dataset was matched with the keywords **covid-19** and **coronavirus**. The regular expressions used for the matching was **r".*covid.{0,5}19.*"**, meaning there could be between 0 and 5 arbitrary characters between **covid** and **19**, and **r".*corona.{0,10}virus.*"**, meaning there could be between 0 and 10 arbitrary characters between **corona** and **virus**. The result clearly showed that domain names containing similar strings to covid-19 and coronavirus saw a spike in submissions on 9:th mars and has since then been on a consistent high level of between 200-350 submissions per day. See figure 1.

In the second test case the database was matched with the keywords **travel**, **hotel**, **health** and **world**. See figure 2. The point was to try and see if there had been a decline in domain registrations in certain branches that might have been badly affected by the coronavirus, e.g. travel and hotel companies. The result showed that indeed there have been a very clear decline in registrations for keywords **travel** and **hotel** on February 7 however, the result showed the same decline for keyword **health** and **world**. Therefore, any such correlation could not be proven. Instead, the results suggested an overall decline of certificates submitted.

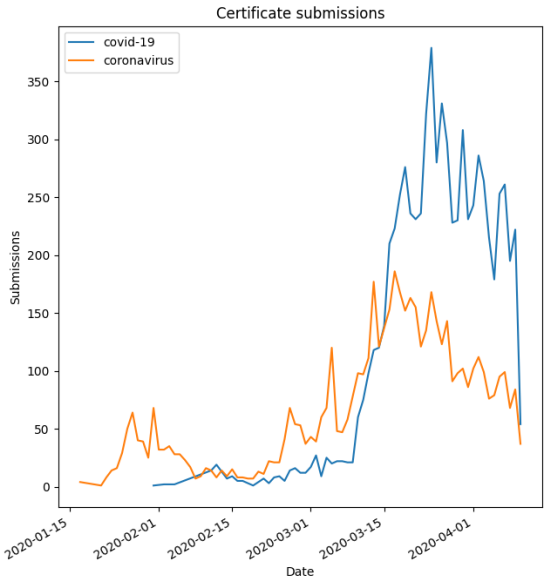

Figure 1. Certificate submissions with keywords covid-19, coronavirus.

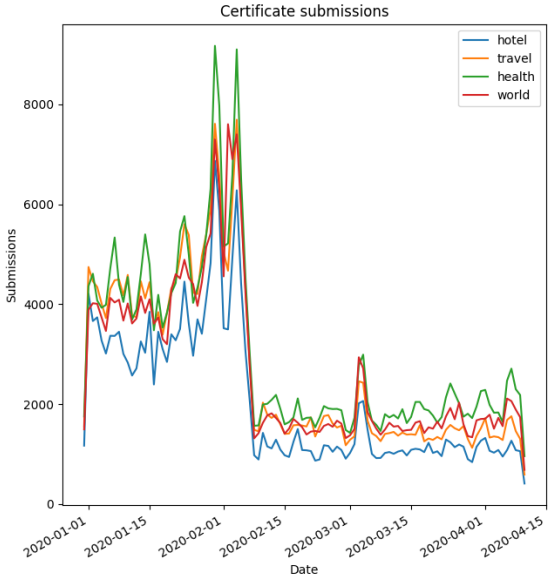

Figure 2. Certificate submissions with keywords travel, hotel, health, world.

4.2 Levenshtein distance checking

Levenshtein distance checking was performed on six authentic domain names. These were chosen because it was believed that they would have attracted more traffic as a result of the coronavirus outbreak. The thesis was that these domain names would also attract scammers attention because of the new popularity of the domains. The condition set for the filtering was if the Levenshtein distance of the domain was less than 25% of the length of the domain string. After the filtering, suspected phishing domains was selected by manual method, i.e. every domain was looked at and chosen manually.

Authentic website	#	Suspected phishing domains
worldometers.info	11	worldometers.live worldmetre.info worldoeters.info worldomters.info worldmoeters.info worldometets.info worldofmeters.info worlmeters.info worldomerers.info worldomaters.info worldmerters.info
healthline.com	5	99healthline.com helth-line.com healhline.com. healthlineed.com healthlibr.com
mayoclinic.org	1	mayoclinnic.org
hopkinsmedicine.org	1	hopkinsmedicare.org
cdc.gov	0	
who.int	0	

Table 1. Subset of suspected phishing domains mimicking authentic websites.

Most of the suspected phishing candidates were either offline or not operational at the time of filtering (2020-05-04). Only one domain, worldometers.live, was on a blocklist and can therefore be safe to say was a phishing domain.

Since then a lot more of these domains have started to become operational as redirects to scam websites luring visitors to buy bitcoin or websites of pornographic content. The majority of domains followed a specific pattern.

1. The certificate the domain belonged to often contained many other suspicious domains, up to 100, many of which were of a pornographic nature or containing keywords like **bitcoin**, **free**, **money** etc.
2. Many of these domains, when visited, had the same generic look and contained the same links. These links often referred to fast easy loans or car insurances. This suggests that the domains are probably not yet operational and could be a part of a future phishing campaign.

5. Discussion/Analysis

The result of the substring matching showed interesting trends and spikes at certain dates. For example, when matching with the keywords **covid-19** and **coronavirus** a clear rise at Mars 9 could be observed. This is interestingly the same day that Italy went into full lockdown. It's also worth noting that WHO named the coronavirus COVID-19 a month earlier at February 11 but no indication of a rise in certificate submissions could be seen around that time. This could maybe be explained with that most people perhaps did not fully grasp the seriousness of the disease until Italy went into lockdown.

The results also showed a strong decline of certain keywords in domain names. Keywords *travel* and *hotel* were expected to have declined during a crisis like this since people stay at home to a greater degree but, keywords *health* and *world* also showed a very similar curvature in the plot so this might not be the case. The results would rather suggest that there has been an overall decline of certificate submissions since February. The reason why the decline happened between 5-8 February is uncertain. Either some real-world event occurred at this date or something might've happened to the CT log that caused this decline. Interesting dates around this period is February 7 when the Wuhan doctor died in coronavirus. The doctor had been trying to warn the public about the deadliness of the virus but was shut down by the Chinese government. This is before WHO named it covid-19 and it was still just called coronavirus.

With the first question at issue in mind the results very clearly showed that yes, there is a correlation between the coronavirus chain of incidents and the registration of domain names with certain keywords.

The Levenshtein method was able to identify a subset of candidates for phishing domains trying to mimic the authentic domain name worldometers.info. The domain healthline.com also returned some potential phishing domains while mayoclinic.org, hopkinsmedicine.org, cdc.gov and who.int did not return a satisfying result. The reason for this is probably because the Levenshtein algorithm is not well suited for finding misspellings of short domain names such as cdc.gov and who.int. If the threshold was too low only authentic certificates of cdc.gov and who.int got through the filtering and if it was too high almost every short domain name would get through.

With the second question at issue in mind the answer is no. The results could not show any trend of phishing domains mimicking authentic websites related to covid-19. The rate of submissions of such malicious certificates were roughly the same before and after the coronavirus outbreak.

In hindsight something more sophisticated than Levenshtein would be needed to get more accurate results. The choice of CT log was another parameter to the poor result. CloudFlare Nimbus 2020 consists only of certificates that have been directly submitted. It would probably be wiser to use Google Pilot CT log which has a web crawler that automatically adds certificates it finds. Therefore, it is probably a higher rate of phishing domains in those types of CT logs.

6. Related work

A lot of inspiration for the methodology came from the good work of Ryan Sears at Cali Dog Security who wrote an article about how to download certificates and import them into Google BigQuery [8]. Ryan Sears is also the author of Axeman that was used in this project.

7. Conclusions

The work consisted of downloading, parsing and storing certificates into a Google BigQuery table. Then with the help of a small python tool the big table was queried with regular expression matching and Levenshtein-distance conditions to find misspellings of authentic domain names. The results were plotted using the python library matlibplot.

The results showed a very clear indication that the coronavirus outbreak have affected the submissions of certificates. Interesting dates were February 7 and Mars 9. On the first date the results show a clear decline of domain names issued containing the keywords **travel** and **hotel** which seem to suggest that these branches have been less profitable during the coronavirus outbreak however, a decline was also present for keywords **health** and **world** which contradicts this suggestion. On the second date the results show a clear rise of domain names containing the keywords **covid-19** and **coronavirus**. February 7 is the date that the Wuhan doctor who was shut down by the Chinese government died in covid-19. Mars 9 is the date Italy went into full lockdown. The conclusion of this is that the CT log landscape is very much affected by the events of the real world.

The Levenshtein approach yielded a subset of potential phishing domains of which only one was blacklisted. The others were either offline or not yet operational, i.e. they contained only some arbitrary links. After revisiting the domains a few days later more had become operational and now acted as redirects to scam websites luring visitors to buy bitcoin. It could not be shown that the frequency of registration of these domains had increased after the coronavirus outbreak but, rather that the frequency was constant. The conclusion of this result is that either the method was flawed, the data wasn't big enough or there simply was no correlation between the coronavirus and registration of phishing domains mimicking popular corona-related websites.

References

[1] Cybersecurity Security [online] available at https://www.varonis.com/blog/cybersecurity-statistics/ [accessed at May 2020]

[2] Transport Layer Security [online] available at https://en.wikipedia.org/wiki/Transport_Layer_Security [accessed at March 2020]

[3] Public key cryptography [online] available at https://en.wikipedia.org/wiki/Public-key_cryptography [accessed at March 2020]

[4] Public key certificate [online] available at https://en.wikipedia.org/wiki/Public_key_certificate [accessed at March 2020]

[5] Diginotar [online] available at https://en.wikipedia.org/wiki/DigiNotar [accessed at May 2020]

[6] What is CT? [online] available at http://www.certificate-transparency.org/what-is-ct [accessed at March 2020]

[7] Hacking web applications before they are installed [online] available at https://www.golem.de/news/certificate-transparency-hacking-web-applications-before-they-are-installed-1707-129172.html [accessed at March 2020]

[8] Retrieving, storing and querying 250m certificates like a boss [online] available at https://medium.com/cali-dog-security/retrieving-storing-and-querying-250m-certificates-like-a-boss-31b1ce2dfcf8 [accessed at April 2020]