The Aftermath of Let's Encrypt's Mass Revocation on IoT Devices

Andrew Chabot
afc1755@rit.edu
Rochester Institute of Technology
Rochester, New York

ABSTRACT

When the largest Certificate Authority, Let's Encrypt, revoked three million certificates on March 4, 2020, users had very little time to renew affected certificates. This is a small timeline for website owners, but the larger issue could be seen in Internet of Things (IoT) device certificates that were revoked in this mass revocation. My study finds that these revoked IoT certificates have not yet been replaced for the devices, leading to millions of potentially vulnerable certificates in the IoT ecosystem.

KEYWORDS

certificates, network security, certificate revocation

ACM Reference Format:
Andrew Chabot. 2020. The Aftermath of Let's Encrypt's Mass Revocation on IoT Devices. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Let's Encrypt was created in 2014 with the goal of democratizing X.509 certificates by providing these certificates at no charge. This is in stark contrast to many other CAs, who charge premiums for businesses to enable TLS encryption for their ecosystems. They have strived towards their stated goal of making encrypted connections ubiquitous by lowering complexity and payment barrier for setting up and maintaining TLS encryption. Based on several other papers, Let's Encrypt has tended to have some of the best marks in many longitudinal studies, from Certificate Transparency Logs to HTTPS security. However, they are not immune from issues and a large example of this happened on the 29th of February, 2020.

On this date, a bug was found with CAA (Certificate Authority Authorization) and the thirty day validation period following a check on CAA records. The bug was such that a subscriber could validate a domain name at a certain time, install invalid CAA records on that domain name, and then still be able to issue certificates with that domain name for thirty days following the initial validation. The bug was quickly fixed later that day, but had been in production since mid 2019. This led to millions of certificates previously being acceptable to be marked as invalid, and Let's Encrypt was forced to revoke all of the affected certificates. In total, approximately 3 million certificates were affected by the bug, around 2.6% of Let's Encrypt's active certificates [2]. This paper will analyze how those revoked certificates have been handled by domain owners. The focal point of this discussion will be looking at re-issuance rates in the most vulnerable of these domains, IoT devices.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 BACKGROUND

I will start by discussing the basics of Certificate Transparency Logs and Let's Encrypt's role in the CT Log ecosystem.

2.1 Certificate Transparency

Certificate Transparency (CT) logs are an append-only system of logs that enable the TLS ecosystem to be auditable. CT was a concept first worked on after popular CA DigiNotar was compromised and the aftermath left fradulent certificates that were very difficult to be spotted and reported. Google and especially the Chrome team became the main proponent of Certificate Transparency over time, and they helped push it into the mainstream very quickly, especially compared to other major internet ecosystem changes. The first log was launched by Google in 2013, and DigiCert became the first CA to implement CT later that year. Extended Validation (EV) Certificates became required to be logged in 2015. In 2018, the largest change came when Chrome began to require CT for all certificates[4].

This requirement led to rapid adoption of the system with potential log overload right at the date of enforcement from Google. As per usual, most of the CAs waited until the last minute to begin logging, potentially because of the privacy concerns that can come with Certificate Transparency. These logs can expose websites and especially subdomains that would normally not be visible to the public. However, over time most have agreed that these tradeoffs have been worth it and that CT has led to companies having much more knowledge about the certificates associated with them.

2.2 Let's Encrypt

Many other companies have since started their own CT logs after Google, with CloudFlare in 2018 and most importantly for this paper, Let's Encrypt's Oak log in 2019. Since then, they have issued more than 1 Million certificates to this log a day and have opened up the log to other trusted Certificate Authorities as well. They also have a monitoring tool created for their CT log called CT Woodpecker that helps to ensure compliance [1]. Most everything that Let's Encrypt was doing in the CT area was ahead of the field or at least in line with the leader, Google. However, lots of negative attention has been thrown their way because of the CAA bug that will be investigated here.

The CAA bug revocation was done over a 24 hour period, giving certificate owners a short window to reissue certificates for their domains before being marked as a security issue. According to Let's

Encrypt, the vast majority of the the affected certificates do not pose any real security threat. However, all of the revoked certificates are not fully compliant and that is enough to give reason for the revocation [2]. An issue that Let's Encrypt was hoping to avoid with this revocation is the concept of "warning blindness". If revoked certificates stay in the field, users will be greeted with a warning from their browser and may just bypass the warning. This wouldn't be a huge issue with the current revoked certificates, but in the future could cause those same users to click through more serious security warnings without thinking.

This sets the stage for the aftermath of the mass certificate revocation. Let's Encrypt is relying on the users of these revoked certificates to reissue in order to minimize warning blindness and keep up their reputation as a reliable Certificate Authority.

3 RELATED WORK

There has been a significant amount of work done in the Certificate Transparency field as a whole. A few papers have focused on adoption of CT as a whole across the ecosystem, but other topics have been looked at as well in the field. Privacy has been a major topic of exploration as well, with exposed subdomains being looked at along with leakage of DNS information throught the logs. CT logs have also been used in attempts to detect phishing domains on a large-scale, with some success [4]. There has also been studies on Certificate Transparency as a model for deployment of ecosystem-wide system change [5].

In terms of prior work with regards to the mass revocation of Let's Encrypt, I was not able to find any papers that had yet breached this topic. I am sure that there are papers in the works focusing on this subject, but none of them seem to have been presented at any sort of conference or been published yet. However, certificate revocation has been looked at many times, including a 2015 paper which checked different browser and operating systems. This paper showed that lots of revocations are ignored by both browsers and users a lot. The user section here helped further the seriousness of warning blindness [3].

4 METHODOLOGY

The CT log data was downloaded from Let's Encrypt's Oak CT log on April 24th, 2020 using the Axeman python utility. The data consists of over 400 Million Certificates that have been issued by Let's Encrypt and other Certificate Authorities. Let's Encrypt made all of the affected certificate serials public, and this data was downloaded from the Let's Encrypt website in order to check against the recent CT log data. In addition to these data sets, I also utilized a script created by Hanno Bock to check if affected hosts had reissued their revoked certificates yet. That code is available here: lecaa. I was not able to download all of the data for the 2020 Oak log, so I have to extrapolate the data based on the results I have for the percentage of the log that I used. I will explain this more in the results section. I also used portions of the 2020 Google Icarus log to combine with this data set. In total, I tested against around 6 million certificates from these various CT logs. The table for certificates tested against by log is below:

From the text file of all affected certificates, I extracted just the serial number using a short Python script. This left over 3 million serials to check using the lecaa script to determine whether the certificates for these domains were still valid. I used the file as input for the lecca script and gathered my results in a text file. I then parsed through the original text file again, looking for domain names that match IoT device configurations versus just normal domains. The code for both of these python scripts can be found here.

Table 1: Certificates by CT Log Used

CT Log	Number Used	% of Log's Certificates
Google Argon 2020	9,898,817	1.86%
LE Oak 2020	17,508,865	4.28%
LE Oak 2021	2,111,155	100%
LE Oak 2022	1,651,680	100%

5 RESULTS

I will preface the results with a disclaimer, since they did not pan out as well as I intended. Unfortunately, I was not able to get full results for each of the logs I looked at and the results basically turned out with me finding no certificates in my datasets from CT logs matching the other dataset. I finished running my script over the Let's Encrypt 2021 log and that may have been a waste of time since that log had mostly test certificates and very newly issued certificates that must have been unaffected by the CAA bug.

The code at my Github can essentially serve as the results here, and I hope to run this code more fully to get more potential results. I am currently running the code on other logs right now with no real results so far, but the scripts will be taking a long time to run. I severely underestimated how much data there were in each of these CT logs, and optimizing these scripts could go a long way in terms of reducing the time it takes to parse through each list and check to see the most updated results. In retrospect, I should have started this over a month ago with the amount of time Axeman takes to download logs and the amount of time it takes to run the scripts I have created. Another possible improvement would have been to run the scripts on a much more powerful computer. As it is, the only computer I have to run Axeman on is my MacBook with two 1.5GHz cores, and that is just not enough power to download hundreds of millions of certificates.

Table 2: Percentage of Certificates for Each Domain

Category	Percentage	Total Number
Valid	100%	2457051(so far)
Invalid	0%	0

Table 3: Breakdown of Invalid Certificates

Category	Percentage of Total Invalid Certificates	Count
IoT	0%	0
Other	0%	0

The results that I do have are below, and I will update them after more of my scripts have finished running in the hope that there will be some actual data to interpret and bring to the discussion.

6 CONCLUSION

The results from looking at all of the revoked certificates from the CAA bug and their status over a month later will hopefully be available soon. Largely, it would be good to see if IoT devices have completely ignored the revoked certificates and have continued as potential security threats in the internet ecosystem. This is what I think would happen, but I would love to have some data to back that up. This lack of re-issuance would be damaging to the CT log system as well as to the reputation of Let's Encrypt. However, it would also be good to see how IoT devices compare in terms of their re-issuance levels. I suspect that the re-issuance rate would be much higher for all of the non-IoT devices.

I do appreciate the ease of access that there is for Certificate Transparency logs. CT logs remain a valuable tool for auditing if certificates are being properly managed, and the usage of CT logs was crucial in order to research a problem such as this.

ACKNOWLEDGMENTS

I would like to thank Professor Taejoong Chung for teaching me many new PKI systems, including CT logs.

REFERENCES

[1] 2019. Introducing Oak, a Free and Open Certificate Transparency Log. https://letsencrypt.org/2019/05/15/introducing-oak-ct-log.html

[2] Josh Aas. 2020. 2020.02.29 CAA Rechecking Bug. https://community.letsencrypt.org/t/2020-02-29-caa-rechecking-bug/114591/3

[3] Yabing Liu, Will Tome, Liang Zhang, David Choffnes, Dave Levin, Bruce Maggs, Alan Mislove, Aaron Schulman, and Christo Wilson. 2015. An End-to-End Measurement of Certificate Revocation in the Webs PKI. *Proceedings of the 2015 ACM Conference on Internet Measurement Conference - IMC 15* (2015). https://doi.org/10.1145/2815675.2815685

[4] T. Nolte J. Amann L. Brent G. Carle R. Holz T. C. Schmidt M. Wählisch Q. Scheitle, O. Gasser. 1998. The Rise of Certificate Transparency and Its Implications on the Internet Ecosystem. (1998).

[5] Emily Stark, Ryan Sleevi, Rijad Muminovic, Devon Obrien, Eran Messeri, Adrienne Porter Felt, Brendan Mcmillion, and Parisa Tabriz. 2019. Does Certificate Transparency Break the Web? Measuring Adoption and Error Rate. *2019 IEEE Symposium on Security and Privacy (SP)* (2019). https://doi.org/10.1109/sp.2019.00027