Human and Organizational Factors in Public Key Certificate Authority Failures

Skyler Johnson
skymjohn@iu.edu
Indiana University Bloomington
Bloomington, Indiana, USA

Katherine Ferro
ktferro@iu.edu
Indiana University Bloomington
Bloomington, Indiana, USA

L. Jean Camp
ljcamp@indiana.edu
Indiana University Bloomington
Bloomington, Indiana, USA

Hilda Hadan
hhadan@iu.edu
Indiana University Bloomington
Bloomington, Indiana, USA

ABSTRACT

Public Key Infrastructure (PKI) is the foundation of secure and trusted transactions across the Internet. Public key certificates are issued and validated by Certificate Authorities (CAs), which have their trust-of-anchor certificates in Root Program Operators' stores. These CAs provide certificates that attest to the integrity of the ownership of domain names on the web and enable secure communications. Each year hundreds of certificates are by these verified and trusted Certificate Authorities issued in error. In this research, we complied and classified certificate incident reports documented on Bugzilla, a web-based bug tracking system where such instances are reported.We focus on the 210 incident reports from the last year; we compare this pandemic period to trends from previous years. Our data show that the frequency of Certificate Authority non-compliance is a consistence source of vulnerability in the PKI ecosystem. The evaluation of reasons for the misissuance illustrate the role of one-off human failures, systematic interaction flaws leading to repeated incidents, and evidence of perverse incentives leading to misissuance.

CCS CONCEPTS

• **Security and privacy** → **Network security**; **Human and societal aspects of security and privacy**.

KEYWORDS

Public Key Infrastructure, Digital Certificate, Certificate Authority, Software Bugs, Non-compliance

ACM Reference Format:
Skyler Johnson, Katherine Ferro, L. Jean Camp, and Hilda Hadan. 2021. Human and Organizational Factors in Public Key Certificate Authority Failures. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security (CCS '21), November 15–19, 2021, Virtual Event, Republic of Korea.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3460120.3485360

1 INTRODUCTION

Misissuance is a chronic problem in the web-based public key infrastructure. Misissuance is distinct from the issuance of rogue certificates, where the issuer is usually judged to either have engaged in malfeasance or been victimized by malicious parties. Nonetheless patterns of misissuance represents a potential vulnerability in the PKI that could be leveraged by attackers.

To determine the ground truth of the causes of flawed certificates as well as the types of failures embedded within them, we compiled reports of incidents from April 2020 to April 2021. Then for each incident we identified and categorized the cause and the type of incident, the party at fault, and public disclosure practices of the entity at fault. An incident may be as small as a single certificate or as large as every certificate from a Certificate Authority (CA). We detail the data from the previous pandemic year, and compare this with trends from the past two decades.

Our results illustrate the role of non-malicious organizational and human failures. We demonstrated how those failures can be seen as resulting primarily for a few problematic CAs and systematic incentive misalignment. We identified and described the most common failures and their causes.

2 MOTIVATION

While there is significant research on developing improved PKI warnings [2, 9, 14], on specific coding errors [1, 7, 12, 13, 16, 17], and on minding your primes [8, 10], there has been less research on the larger patterns of incidents in the CA ecosystem as a whole.

An early survey that highlighted systematic problems in CA practices, particularly the existence of digital certificates not compliant with the requirements of the CA/Browser Forum, was completed by researchers at INRIA and Microsoft [3]. Later, Kumar et al. showed that some CAs have been issuing erroneous digital certificates since their earliest days of operations [11]. Gasser er al. compiled certificates from scanning the network by querying Certificate Transparency logs, finding numerous mis-configured digital certificates that had been in active use [5]. In 2016, Dong et al. [4] proposed decentralized local machine-learning to identify unfamiliar and potentially malicious certificates because, like Gasser, they found invalid facts and flawed cryptography; further their targeted analysis showed that these were more common in depository institutions than average.

CCS '21, November 15–19, 2021, Virtual Event, Republic of Korea.

ACM ISBN 978-1-4503-8454-4/21/11.
https://doi.org/10.1145/3460120.3485360

Our work placed these evaluations in the larger context by identifying and classifying the root causes. In the next section, we describe our data sources and analysis method. We then detail the failures found in the past year; after which we place these results in the context of failures in the past two decades. We conclude with a discussion of evidence of incentive misalignment in the public key infrastructure as a whole.

2.1 Data Compilation and Analysis Method

We compiled a comprehensive dataset of public key certificate incidents. For each incident, we recorded the date, primary cause, the scope, the reporting entity, and the associated CA. From this we provide a comprehensive data analysis to identify major types of PKI incidents, major PKI offenders, public disclosure practices, and major types of causes.

We chose data sources that were public, consistent, impartial, and trustworthy. Sources with consistent syntax did not require parsing and pre-coding. The core of our incident collection was Mozilla's Bugzilla[1]. This source met the requirements and offers a database of public incidents related to PKI with consistent syntax in the reporting structure. The resulting 210 incidents occurred between April 2020 to April 2021.

Quantitative Data Compilation: We began by classifying the incidents with clear unambiguous data. Information that was consistently reported included year, entity, Root CA, and disclosing party. Incident cause and type often required additional coding.

Year refers to the year in which the incident was reported. Generally this is the year the incident happened; however, some where incidents spanned multiple years or were reported retrospectively. *Entity* refers to entity erred in issuance. These include CAs, Intermediate CAs, Resellers, Registration Authorities, or Auditors. The *Root CA* for each incident is Root CA whose is the base of the chain of trust. This will be an entity whose root digital certificates are included in the Root Programs. *Disclosing party* is a Boolean variable identifying if incident was disclosed by the responsible entity or Root CA. *The type of incident* that was identified in the disclosure often required qualitative coding, but was in some cases directly noted in the disclosure. In some instances the *the cause of the incident* was also available. The causes of the instances required further analysis.

Cause	# of incidents	# of self-reported incidents	self-report rate (%)
Software bugs	34	17	50.00%
Single cert human error	30	11	36.67%
Requirement 'unknown'	20	4	20.00%
CA business decision	15	5	33.33%
Operational error	84	36	42.86%
Change in Baseline Requirements	6	1	16.67%
Organizational constraints	6	2	33.33%
Improper security controls	1	0	0%
Non-optimal request check	1	0	0%
Other	15	7	46.67%
No data	13	2	15.38%
Total	210	59	28.10%

Table 1: ***The name of the cause; then for each cause the number of total incidents, the number of incidents reported by the CA, and percentage of CA self-reported incidents.***

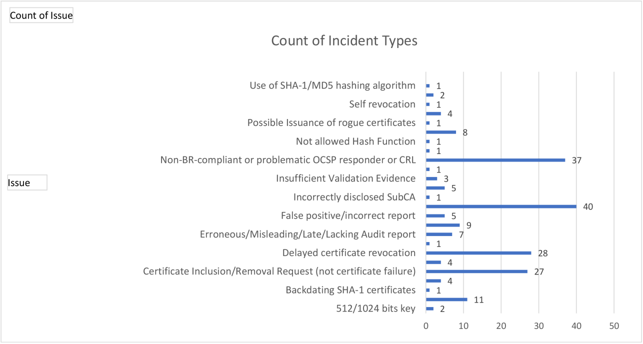
Root Cause Classification: We conducted a qualitative analysis to discover the causes of incidents. We used a codebook developed by the two primary researchers who read 1,800 incident reports dating from 2001 to March 2020. The codebook was then evaluated by a focus group of eight graduate researchers using randomly selected incidents to determine if the codebook was usable, accurate, and understandable. This review further evaluated the codebook by having reviewer with different research perspectives and levels of expertise [15] As a final confirmation of the coding, one undergraduate computer science researcher and one doctoral researcher independently read and classified each incident report between April 2020 to April 2021. Any cases of inter-coder disagreements or lack of clear fit with causes of the codebook were discussed and resolved at weekly meetings. Every incident has been read and classified between two to five times, with every incident having at least one identified cause. Multiple incidents have a set of causes, in which case coders identified a dominant cause. Events before April 2020 have all been classified at least five times.

The results of the qualitative data analysis and the incident features provide meta-data that identifies trends in incidents, problematic entities, the most common types of incidents, the causes of incidents, disclosure practices, and yearly comparisons for each of these factors.

3 CONTRIBUTION

Our results illustrated that there is an increase in the number of incidents where CAs violated the Baseline Agreements during the pandemic year. In the past year both the number of incidents and the number of CAs associated with at least one incident, increased. Despite this increase, for the vast majority of trusted CAs no incident was reported. We identified the erring entities, the ways in which they have failed, and the trends of behavior among CAs. It is difficult to avoid the conclusion that ignoring the Baseline Requirements is increasing, and that CAs can do so without risk of costly consequences.

Figure 1: ***The chart below shows the distribution of types of incidents between April 2020 to April 2021.***



We consider the data in context of analysis of previous year. We identify inflection points in historical data that correspond to significant changes in PKI. The rollout of Certificate Transparency correlates in an immediate increase in discovery of incidents, then increase in third-party reporting, and in the next year there an increase in self-reports.

[1] https://bugzilla.mozilla.org

Figure 2: ***The chart below shows the distribution of primary causes of incidents between April 2020 to April 2021.***

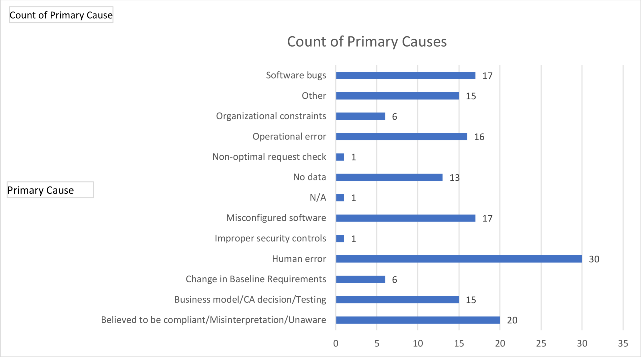
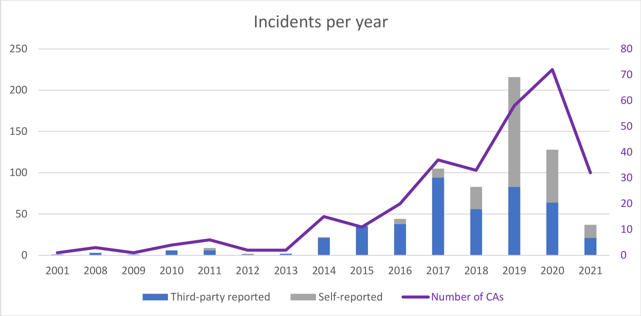


Figure 3: ***The chart below shows the number of self-reported incidents, and the number of CAs involved are shown in dash-lines. Note the increase in both incidents and the number of CAs associated with an incident to 2019; then an increase in percentage of self-reports and a drop in incidents in 2020 and thus far in 2021.***



Based on reports from a previous workshop, interviews reported in [6], and in fact, our own preconceived notions the frequency of Certificate Authority non-compliance is a greater source of vulnerability than generally discussed. Non-compliance is not only from rogue certificates issued by malicious or hacked CAs, but rather result from standard operating procedures. For example, signing certificates with keys that will expire in the certificate lifetime generates browser warnings, could be reported, appeared to be a not uncommon business practice, and arguably resulted from incentive misalignment.

One limitation was that our dataset addressed only reported incidents. In addition, impact was difficult to evaluate even when number of certificates is available. For example, in spring of 2020 an expired Comodo signing certificate blocked new donors from ActBlue during Democratic primaries.

As the interaction space for warnings and indicators decreased with IoT and embedded systems, while the potential harm for failure increases, the advances in transparency on for web certificates is even more critical.

ACKNOWLEDGMENTS

This research was supported in part by the National Science Foundation under CNS 1565375, Cisco Research Support, and Comcast Innovation Fund. Any opinions, findings, and conclusions expressed in this material are those of the author(s) and do not necessarily reflect the views of the the US Government, IU, NSF, nor any funding entity. We would like to acknowledge past contributions by Nicolas Serrano, Olivia Kenny, Isaac Streiff, Jack Ruocco, Trevor Cutshall,and Joshua Johnson.

REFERENCES

[1] Yasemin Acar, Sascha Fahl, and Michelle L Mazurek. 2016. You are Not Your Developer, Either: A Research Agenda for Usable Security and Privacy Research Beyond End Users. In *2016 IEEE Cybersecurity Development (SecDev)*. IEEE, 3–8.

[2] Bonnie Brinton Anderson, C Brock Kirwan, Jeffrey L Jenkins, David Eargle, Seth Howard, and Anthony Vance. 2015. How Polymorphic Warnings Reduce Habituation in the Brain: Insights from an fMRI Study. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 2883–2892.

[3] Antoine Delignat-Lavaud, Martin Abadí, Matthew Birrell, Ilya Mironov, Ted Wobber, and Yinglian Xie. 2014. Web PKI: Closing the Gap between Guidelines and Practices. In *Proceedings of the ISOC Network and Distributed System Security Symposium (NDSS ’14)*. http://antoine.delignat-lavaud.fr/doc/ndss14.pdf

[4] Zheng Dong, Kevin Kane, and L. Jean Camp. 2016. Detection of Rogue Certificates from Trusted Certificate Authorities Using Deep Neural Networks. *ACM Transactions on Privacy and Security* 19, 2 (Sep 2016), 1–31.

[5] Oliver Gasser, Benjamin Hof, Max Helm, Maciej Korczynski, Ralph Holz, and Georg Carle. 2018. In Log We Trust: Revealing Poor Security Practices with Certificate Transparency Logs and Internet Measurements. In *Passive and Active Measurement*. Springer International Publishing, 173–185.

[6] Hilda Hadan, Nicolas Serrano, Sanchari Das, and L Jean Camp. 2019. Making IoT Worthy of Human Trust. *Available at SSRN 3426871* (2019).

[7] Michael P. Heinl, Alexander Giehl, Norbert Wiedermann, Sven Plaga, and Frank Kargl. 2019. MERCAT: A Metric for the Evaluation and Reconsideration of Certificate Authority Trustworthiness. In *Proceedings of the 2019 ACM SIGSAC Conference on Cloud Computing Security Workshop* (London, United Kingdom) *(CCSW’19)*. Association for Computing Machinery, New York, NY, USA, 1–15. https://doi.org/10.1145/3338466.3358917

[8] Nadia Heninger, Zakir Durumeric, Eric Wustrow, and J. Alex Halderman. 2012. Mining Your Ps and Qs: Detection of Widespread Weak Keys in Network Devices. In *Presented as part of the 21st USENIX Security Symposium (USENIX Security 12)*. USENIX, Bellevue, WA, 205–220. https://www.usenix.org/conference/usenixsecurity12/technical-sessions/presentation/heninger

[9] Katiana Krawchenko. 2016. The Phishing Email That Hacked the Account of John Podesta. https://www.cbsnews.com/news/the-phishing-email-that-hacked-the-account-of-john-podesta/.

[10] Katharina Krombholz, Karoline Busse, Katharina Pfeffer, Matthew Smith, and Emanuel von Zezschwitz. 2019. "If HTTPS Were Secure, I Wouldn’t Need 2FA" - End User and Administrator Mental Models of HTTPS. In *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 246–263.

[11] Deepak Kumar, Zhengping Wang, Matthew Hyder, Joseph Dickinson, Gabrielle Beck, David Adrian, Joshua Mason, Zakir Durumeric, J. Alex Halderman, and Michael Bailey. 2018. Tracking Certificate Misissuance in the Wild. In *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE.

[12] Adam Langley. 2014. Apple’s SSL/TLS Bug. https://www.imperialviolet.org/2014/02/22/applebug.html.

[13] Microsoft. 2020. CVE-2020-0601 | Windows CryptoAPI Spoofing Vulnerability. https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2020-0601.

[14] Robert W. Reeder, Adrienne Porter Felt, Sunny Consolvo, Nathan Malkin, Christopher Thompson, and Serge Egelman. 2018. An Experience Sampling Study of User Reactions to Browser Warnings in the Field. In *CHI Systems*. ACM, New York, NY, USA.

[15] Johnny Saldana. 2017. The coding manual for qualitative researchers. *Qualitative research in organizations and management: an international journal* (2017).

[16] Synopsys Editorial Team. 2014. Understanding the Apple ’goto fail;’ Vulnerability. https://www.synopsys.com/blogs/software-security/understanding-apple-goto-fail-vulnerability-2/.

[17] David A. Wheeler. 2017. The Apple Goto Fail Vulnerability: Lessons Learned. https://dwheeler.com/essays/apple-goto-fail.html.