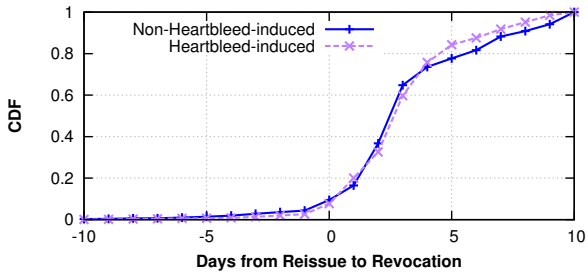

Figure 14: Cumulative distribution of the number of days between when a certificate is reissued and when it is revoked. Positive values indicate the certificate is reissued before it is revoked; negative values indicate the opposite.

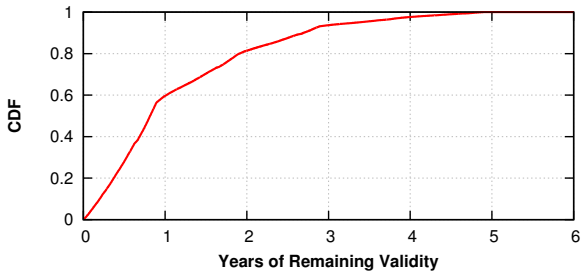

Figure 15: The distribution of time-until-expiry for vulnerable, reissued, but not revoked certificates. If these certificates are never revoked, this figure shows how long they will persist.

sues, i.e., sites with high rank are slightly more likely to revoke. Ideally, the two lines in Figure 10 should be coincident, i.e., all sites reissuing certificates due to Heartbleed should also have revoked the retired certificates (the only exception to this rule is if the retired certificate was about to expire anyway, but we account for this in our definitions of Heartbleed-induced reissues and revocations). This result highlights a serious gap in security best-practices across all of the sites in the Alexa Top-1M.

Finally, we examine the *revocation speed*, or the number of days between when a certificate is reissued and it is revoked. Figure 14 presents the cumulative distribution of the revocation speed for both Heartbleed-induced and non-Heartbleed-induced revocations. To make the distributions comparable, we only look at differences between -10 and 10 days (recall that Heartbleed-induced reissues and revocations can only occur after April 7, 2014, limiting that distribution). We observe that Heartbleed-induced revocations appear to happen slightly more quickly, thought not to the extent one might expect, given the urgent nature of the vulnerability. We also observe that revocation almost always happens *after* reissue, which is likely explained by the more manual process that revocation often entails. This result contradicts previous assumptions [8] that revocations and reissues occur simultaneously. Finally, it is worth noting that the granularity of our scans makes generalizing these results difficult, since we cannot tell exactly when a certificate was reissued; however, the two distributions are comparable to each other.

Expirations are not enough. To demonstrate how long the effects of this vulnerability could be felt if sites do not revoke their vulnerable certificates, we analyze certificates that, by the end of our data collection, were found to be vulnerable (and alive) when Heartbleed was announced, reissued thereafter, but never revoked. Figure 15 presents the distribution of how much longer such certificates will continue to live if their sites do not revoke them. Note that this CDF appears to be piecewise linear at intervals of 1 year: this is because expiration dates are typically set at intervals of a year—that the distribution is roughly uniform within these year intervals indicates that certificates are issued mostly uniformly throughout the year. This figure shows that, without revoking, the vulnerability introduced in 2014 could affect clients through 2020. We conclude from this that, given the meager rates of revocation, it would be helpful for CAs to shift to shorter expiry times in their certificates.

CRL reason codes. The CRL specification allows the maintainers of CRLs to include a *reason* for why a certificate was revoked along with the revocation in the form of a small set of *reason codes*. The reason code is optional, and the options range from "Unspecified" to "Key Compromised" to "Privilege Withdrawn" [6]. Note that the CRL reason codes are not necessarily verified by the certificate authorities, and they may be incorrect.

For all of the certificates that we observed to be revoked, we extracted the reason code (if one existed); we present the distribution of these reason codes for both Heartbleed-induced and non-Heartbleed-induced certificate reissues in Figure 16. Note the log-scale on the x-axis.

We make two key observations. *First*, we see a significant increase in the probability of a reason code being provided at all for Heartbleed-induced revocations: only 19.2% of non-Heartbleed-induced revocations provide any reason code (*including* the "Unspecified" reason code), while 27.1% on Heartbleed-induced revocations provide a reason code. *Second*, we observe a large increase in the "Key Compromise" reason code (from 0.40% to 1.18% of all CRL entries); given that Heartbleed certificates are likely being reissued

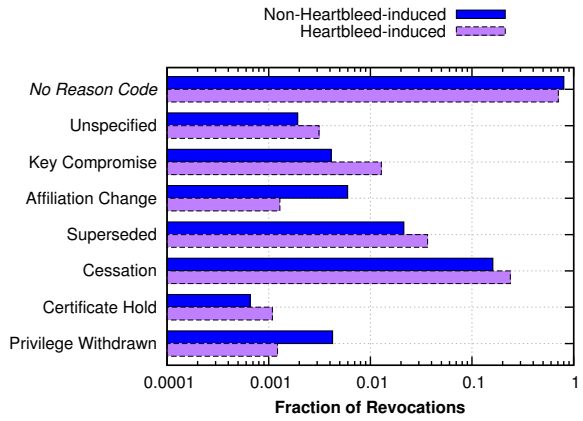

Figure 16: Distribution of CRL reason codes given for both Heartbleed-induced and non-Heartbleed-induced certificate reissues. Note the log scale on the x-axis. We observe an increase in reasons for revocations being given for Heartbleed-induced reissues, especially for the "Key Compromised" reason code.

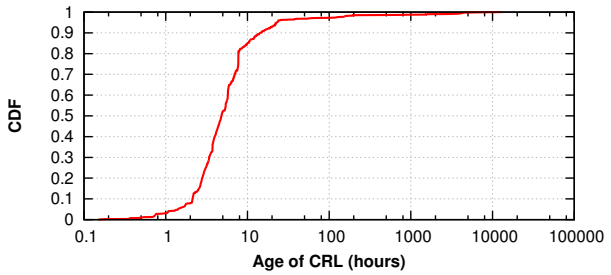

Figure 17: Cumulative distribution of the time between when we downloaded the CRLs (6:00pm EST) and the time of issue recorded in the CRL (and signed by the CA). Most CAs have a chance to revoke certificates at least once a day, as 95% of the CAs updated their CRLs within 24 hours of when we downloaded them.

due to concerns that the private key may have been compromised, this increase is not unexpected. However, it still appears that vast majority of CRL entries are mis-coded. Prior work has also noted that CRLs are usually mis-coded [8], although the snapshot we present in Figure 16 is even more stark, given that we know Heartbleed-induced revocations should have been revoked with a reason code of "Key Compromise".

CRL update intervals. The general lack of site administrators revoking certificates when they should (e.g., after Heartbleed) could be attributed to the CAs only updating their CRLs on very long timescales. For example, one reason for this would be if CAs kept their private keys on offline hosts that would have to be powered on every time to sign CRLs. Another reason would be so clients do not need to download new CRLs very often.

Figure 17 indicates that neither of these reasons are true. This figure shows the cumulative distribution of the difference between the time we downloaded a CRL and the time it was issued. We see that 95% of CAs signed a fresh CRL within 24 hours of 6:00pm EST (when we downloaded the CRLs). When CAs sign a fresh CRL, they have the opportunity to revoke more certificates. These results suggest that CAs could revoke certificates as often as every few hours. Thus, any delays in the revocation of certificates are due to humans in the loop: either certificate owners who are not reporting potentially compromised keys, or CA personnel who are not manually adding new entries to CRLs before they are signed and shipped.

Another important factor in the context of client impact is when (and whether) clients obtained the list of revocations. Unfortunately, we are unable to answer this question given our data collection methodology (it would require instrumenting end-hosts to see when precisely their browsers and operating systems fetched CRLs or issued OCSP queries). Such a study is an interesting area of future work. However, there is one aspect of this problem to which we may be able to lend insight; it was recently reported that many browsers do not even bother to check certificates' CRLs, with the exception of extended validation (EV) certificates [7]. We next turn to an analysis of how these EV certificates are reissued and revoked in comparison to the entire corpus of certificates.

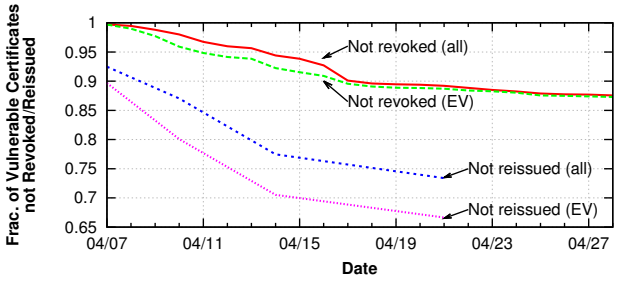

Figure 18: The rate at which vulnerable certificates were reissued and revoked after Heartbleed's announcement. (Note that the y-axis does not begin at zero.)

4.5 Extended Validation Certificates

Recall that one of the major roles of a CA is to validate the identity of the subjects for whom it issues certificates. *Extended Validation* (EV) certificates are a means by which CAs can express that this identity-verification process has followed a set of (presumably stringent) established criteria. EV certificates are standard X.509 certificates, and offer no additional security per se, but the rationale is that with a more thorough verification process by the CAs, these certificates can be more readily verified and trusted by users.[8] That said, there remains concern as to whether or not this trust is well-placed. We close this section by investigating the rate at which vulnerable EV certificates were revoked and reissued as compared to the entire aggregate of certificates.

Figure 18 shows the fraction of vulnerable certificates that have yet to be reissued or revoked over time. In this figure, the initial y values do not all start at 1.0 for reissues: this is because, with coarse granularity of our data, we cannot be certain whether some certificates were reissued immediately after the scan on April 7, 2014, immediately before the scan on April 10, 2014, or in between. We therefore provide the *most optimistic* possibility: if we know a certificate was reissued between days d and $d + k$, then we plot it as having been reissued on day d. The coarse granularity of the scans also explains why the reissue lines do not advance beyond April 21.

Regardless, one trend that remains clear is that sites are more proactive in reissuing new certificates than in revoking old ones. This contradicts prior assumptions that revocations and reissues occur simultaneously [8]. Indeed, it is not yet clear to us *why* a site would reissue a vulnerable certificate without revoking it, but these trends demonstrate that it is a common practice, even for those with EV certificates.

This figure shows a generally bleak view of how *thoroughly* sites revoke and reissue their certificates when necessary. Note that the y-axis begins at 0.65: three weeks after the revelation of Heartbleed, over 87% of all certificates we found to be vulnerable have yet to be revoked, and over 73% of them have yet to be reissued. Of those that *did* revoke their certificates, we find that the speed at which they did so matches that of earlier studies on the spread of patches [25, 27]: there is an exponential drop-off, followed by a gradual decline. Specifically, the "Not revoked (all)" line fits the

[8]Many browsers present EV certificates with a green box in the address bar, while non-EV certificates are often just represented with a gray lock icon.

curve $0.179e^{-0.073x} + 0.830$, while the "Not revoked (EV)" line fits the curve $0.144e^{-0.118x} + 0.859$.

Overall, EV certificates follow similar trends to the entire corpus, with a slightly faster and more thorough response. Interestingly, while EV certificates were revoked more quickly, their non-EV counterparts caught up within ten days; however, EV certificates were reissued both more quickly and more thoroughly. We expect that the underlying cause of this observation is a self-selection effect, i.e., security-conscious sites are more likely to seek out EV certificates in the first place. We doubt that the additional identity verification steps required to obtain an EV certificate play a large role in this (slightly) improved reaction to Heartbleed. Nonetheless, there are still many vulnerable EV certificates that have not been reissued two weeks after the event (67%) and that have not been revoked three weeks after (87%).

5. RELATED WORK

Our work lies at the intersection of two general areas of prior work: studies of how effectively administrators react to widely publicized vulnerabilities, and measurements of the TLS/SSL certificate ecosystem. To the best of our knowledge, we are the first to look specifically at how potentially compromised certificates are replaced and revoked.

Vulnerability patching. There have been several studies of how quickly and effectively administrators patch well-known software vulnerabilities. Rescorla measured the response to a 2002 buffer overflow vulnerability in OpenSSL [27], and Ramos investigated how the fraction of vulnerable systems changes after various security holes from 2000–2005 had been published [25]. Both of these studies found an exponential decrease in the fraction of vulnerable hosts shortly after public revelation of the vulnerability, followed by a gradual decline thereafter. Interestingly, in Rescorla's study, another sharp decline in the number of vulnerable hosts occurred after the release of the Slapper worm which exploited the buffer overflow.

Closely related to our study is that of Yilek et al., who measured the aftermath of a 2008 vulnerability in Debian's OpenSSL key generation that resulted in predictable RSA keys [35]. What makes this work particularly related to ours is that fixing the vulnerability required not only patching OpenSSL, but also reissuing new keys. They found that this process resulted in a gradual decline in the fraction of vulnerable hosts, as opposed to the sharp exponential decay when only patching the software is necessary. However, because their data collection only began several days after the vulnerability was released, the sharp decline may have occurred but gone unnoticed. Our data covers months leading up to and weeks after Heartbleed, allowing us more confidence in the initial drop-off of vulnerabilities.

Our work broadly builds on these prior studies in that we focus on a different, though equally important, aspect of the vulnerability fixing cycle: when potentially compromised certificates were not only replaced, but explicitly *revoked*. The connection between patching software, reissuing new certificates, and revoking old ones has, to the best of our knowledge, not been explicitly studied. Though it had been previously believed that revocations and reissues occur simultaneously [8], our results demonstrate that revocations are often offset in time, or simply never occur at all.

The certificate ecosystem. In focusing on vulnerability fixing as it pertains to certificates, our work is also related to recent studies of the certificate ecosystem at large. Holz et al. [17] performed passive and active measurements on HTTPS certificates from the Alexa Top-1M domains. Durumeric et al. [8] performed active measurements using ZMap [10] that yielded nearly 40× more certificates than prior studies [11, 16, 17]. Broadly, these studies exposed several grim properties of today's certificate ecosystem, including weaker key lengths than suggested by NIST [3], longer certificate chains than necessary, invalid subject names, and so on. Comparing these studies to one another, it appears that the Alexa Top-1M sites—though still far from perfect—do manage certificates more appropriately on average, with a slight weight to higher-ranked domains. Like Holz et al., our work focuses solely on the Alexa Top-1M; we expect that expanding to more domains would, as Durumeric et al. found [8], result in less effective certificate management, though this is an area of future work.

While these studies have shed considerable light on the certificate ecosystem (and found it to be surprisingly bleak), our study is the first to explicitly consider reissues and revocations, particularly in the wake of a widespread vulnerability. Durumeric et al. [8] briefly investigated certificate revocations, and found that a mere 2.5% of the certificates they encountered were ever revoked—of these, the majority gave no reason code. By using Heartbleed as a wide-scale correlated event, we complement this prior work by investigating which certificates *should* have been revoked, and *when* the revocations should have taken place. In the context of the certificate ecosystem, we believe this to be novel.

Heartbleed. The recent nature of the Heartbleed vulnerability means little scientific work has yet to come out studying the vulnerability itself and the community's reaction to it. The most closely related work—a study performed concurrently with our own—presents a comprehensive study of the breadth of the vulnerability, the clean-up, and surveys of administrators who failed to patch their servers [9]. Interestingly, the study leverages historic packet traces [19] to look for evidence of Heartbleed exploitation *before* the announcement and finds no evidence that the vulnerability was exploited beforehand. This study and our own are complementary—theirs briefly examines SSL certificate reissues and revocations, and the results of their analysis are in agreement with ours.

6. CONCLUDING DISCUSSION

In this paper, we study how SSL certificates are reissued and revoked in response to a widespread vulnerability, Heartbleed, that enabled undetectable key compromise. We conducted large-scale measurements and developed new methodologies and heuristics to determine how the most popular 1 million web sites reacted to this vulnerability in terms of certificate management, and how this impacts security for clients that use them.

We found that the vast majority of vulnerable certificates have not been reissued; further, of those domains that reissued certificates in response to Heartbleed, 60% do not revoke their vulnerable certificates—if they do not eventually become revoked, 20% of those certificates will remain valid (not expire) for two or more years. The ramifications of this findings are alarming: modern Web browsers will re-

main potentially vulnerable to malicious third parties using stolen keys to masquerade as a compromised site for a long time to come. We analyzed these trends with vulnerable EV certificates, as well, and have found that, while they exhibit better security practices, they still remain largely not reissued (67%) and not revoked (88%) even weeks after the vulnerability was made public.

To the best of our knowledge, our focused study on certificate reissues and revocations is the first of its kind. Our results are, in some ways, in line with previous studies on the rates at which administrators patched vulnerable software—for instance, revocation rates followed a sharp exponential drop-off shortly after the vulnerability was made public, and tapered off relatively soon thereafter. However, unlike with software patches, we find the vast majority of certificates have still not been reissued or revoked. These findings indicate quite simply that the current practices of certificate management are misaligned with what is necessary to ensure a secure PKI.

Surveying system administrators. To help better understand the reasons behind the lack of prompt certificate reissues and revocations, we informally surveyed a few systems administrators. We asked what steps they had taken in response to Heartbleed: did they patch, reissue, and revoke, and if not, then why not? We received seven responses. Most reported patching their systems, typically in direct response, but some relied on managed servers or automatic updates and therefore took no Heartbleed-specific steps. There was some variance in when patches were applied, due to a combination of scheduled reboots and delayed responses from some vendors, but the majority of patches were applied quickly.

For revoking and reissuing, however, we saw a wide spectrum of behavior. Few both revoked and reissued, but among them, they did so within 48 hours. Many neither revoked nor reissued; a common reason provided was that the vulnerable hosts were either not hosting sensitive data or were not running services that were deemed sensitive enough to warrant it. Along similar reasons, others reported having reissued the certificate but not revoking, explaining that the certificate is only for internal use. Finally, others reported that they did not perceive reissuing and revoking as important because they had patched quickly after the bug was publicly announced (recall, however, that the vulnerability was introduced over two years prior).

Our results from this small survey should be viewed anecdotally—a more extensive survey on certificate administration is an interesting area of future work—but they do shed light on some of the root causes of why revoking and reissuing are not on equal footing with patching. While administrators almost universally understand the importance of patching after a vulnerability, many do not appreciate or know about the importance of revoking and reissuing certificates with new keys. Of those administrators who do understand the importance, even some of them reported push-back from others who perceived the process as being overly complex. In sum, this points to the need for broader education on the treatment of certificates, and perhaps more assistance from CAs to help ensure that all the prescribed steps are taken.

Lessons learned. Our results suggest several changes to common PKI practices that may improve security in practice. First, the practices of low revocation rates and long expiration dates form a dangerous combination. Techniques that automate revocation would vastly reduce the period during which clients are vulnerable to malicious third parties. Similarly, setting reasonably short certificate expiration dates (as suggested by Topalovic et al. [34]) by default will significantly reduce the period during which vulnerable certificates are valid. Second, mechanisms that enable a simultaneous reissue-and-revoke for a certificate will make it less likely that invalid certificates are accepted by clients. Third, we have found that many domains, when they reissue a certificate, continue to offer the old, vulnerable certificate, as well. Given the large number of certificates and hosts using them per domain in our dataset, we believe administrators would benefit from tools that more easily track and validate the set of certificates they are using.

Future work. This paper is, we believe, the first step towards understanding the manual process of reissuing and revoking certificates in the wake of a vulnerability. Several interesting open problems remain. Because our data focuses on the server and CA side of the PKI ecosystem, we are unable to draw any direct conclusions as to what clients experience. A host-centered measurement study would, for instance, allow us to understand not only when revocations were added to CRLs, but when clients actually received the CRLs. Moreover, our study opens many questions as to *why* the certificate reissue and revocation processes are so extensively mismanaged. Our results reinforce previous findings that site popularity is correlated with good security practices, but even the highest ranked Alexa websites show relatively anemic rates of reissues and revocations. Understanding the root causes is an important step towards developing secure infrastructures that effectively incorporate (or mitigate) the end-user administrators.

Open source. Our analysis relied on both existing, public sources of data and those we collected ourselves. We make all of our data and our analysis code available to the research community at

`https://ssl-research.ccs.neu.edu`

Acknowledgments

We thank the anonymous reviewers and our shepherd, Jelena Mirkovic, for their helpful comments. We also thank Rapid7 for collecting the SSL certificate data, the authors of ZMap for collecting the Heartbleed vulnerability data, and for making it publicly available. Finally, we thank our survey respondents for their candid responses. This research was supported in part by NSF grants CNS-1054233, CNS-1319019, and CNS-1150177, and an Amazon Web Services in Education grant.

7. REFERENCES

[1] D. E. 3rd. Transport Layer Security (TLS) Extensions: Extension Definitions, Jan. 2011. IETF RFC-6066.

[2] Alexa Top 1 Million Domains. `http://s3.amazonaws.com/alexa-static/top-1m.csv.zip`.

[3] E. Barker, W. Barker, W. Burr, W. Polk, and M. Smid. Recommendation for Key Management – Part 1: General (Revision 3), 2012. NIST Special Publication 800-57.

[4] Botan SSL Library. http://botan.randombit.net.

[5] CERT Vulnerability Note VU#720951: OpenSSL TLS heartbeat extension read overflow discloses sensitive information. http://www.kb.cert.org/vuls/id/720951.

[6] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. IETF RFC-5280, May 2008.

[7] R. Duncan. How certificate revocation (doesn't) work in practice, 2013. http://news.netcraft.com/archives/2013/05/13/how-certificate-revocation-doesnt-work-in-practice.html.

[8] Z. Durumeric, J. Kasten, M. Bailey, and J. A. Halderman. Analysis of the HTTPS certificate ecosystem. In *ACM Internet Measurement Conference (IMC)*, 2013.

[9] Z. Durumeric, J. Kasten, F. Li, J. Amann, J. Beekman, M. Payer, N. Weaver, J. A. Halderman, V. Paxson, and M. Bailey. The matter of Heartbleed. In *ACM Internet Measurement Conference (IMC)*, 2014.

[10] Z. Durumeric, E. Wustrow, and J. A. Halderman. ZMap: Fast Internet-wide Scanning and Its Security Applications. In *USENIX Security Symposium*, 2013.

[11] P. Eckersley and J. Burns. An observatory for the SSLiverse. In *Defcon 18*, 2010. https://www.eff.org/files/DefconSSLiverse.pdf.

[12] F. F. Elwailly, C. Gentry, and Z. Ramzan. QuasiModo: Efficient certificate validation and revocation. In *Public Key Cryptography (PKC)*, 2004.

[13] P. Evans. Heartbleed bug: RCMP asked Revenue Canada to delay news of SIN thefts, 2014. http://www.cbc.ca/news/business/heartbleed-bug-rcmp-asked-revenue-canada-to-delay-news-of-sin-thefts-1.2609192.

[14] Faketime library. http://www.code-wizards.com/projects/libfaketime/.

[15] B. Grubb. Heartbleed disclosure timeline: who knew what and when, 2014. http://www.smh.com.au/it-pro/security-it/heartbleed-disclosure-timeline-who-knew-what-and-when-20140415-zqurk.html.

[16] N. Heninger, Z. Durumeric, E. Wustrow, and J. A. Halderman. Mining your Ps and Qs: Detection of widespread weak keys. In *USENIX Security Symposium*, 2012.

[17] R. Holz, L. Braun, N. Kammenhuber, and G. Carle. The SSL landscape – A thorough analysis of the X.509 PKI using active and passive measurements. In *ACM Internet Measurement Conference (IMC)*, 2011.

[18] Revocation doesn't work. https://www.imperialviolet.org/2011/03/18/revocation.html.

[19] S. Kornexl, V. Paxson, H. Dreger, A. Feldmann, and R. Sommer. Building a time machine for efficient recording and retrieval of high-volume network traffic. In *ACM Internet Measurement Conference (IMC)*, 2005.

[20] Mac OS X 10.9.2 Root Certificates. http://support.apple.com/kb/HT6005.

[21] S. Micali. NOVOMODO: Scalable certificate validation and simplified PKI management. In *PKI Research Workshop*, 2002.

[22] P. Mutton. Half a million widely trusted websites vulnerable to heartbleed bug, 2014. http://news.netcraft.com/archives/2014/04/08/half-a-million-widely-trusted-websites-vulnerable-to-heartbleed-bug.html.

[23] M. Naor and K. Nissim. Certificate revocation and certificate update. In *USENIX Security Symposium*, 1998.

[24] OpenSSL Project. https://www.openssl.org.

[25] T. Ramos. The laws of vulnerabilities. In *RSA Conference*, 2006. http://www.qualys.com/docs/Laws-Presentation.pdf.

[26] Rapid7 SSL Certificate Scans. https://scans.io/study/sonar.ssl.

[27] E. Rescorla. Security holes... Who cares? In *USENIX Security Symposium*, 2003.

[28] R. L. Rivest. Can we eliminate certificate revocation lists? In *Financial Cryptography (FC)*, 1998.

[29] S. Santesson, M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams. X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP, June 2013. IETF RFC-6960.

[30] R. Seggelmann, M. Tuexen, and M. Williams. Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS) Heartbeat Extension, Feb. 2012. IETF RFC-6520.

[31] N. Sullivan. The Heartbleed Aftermath: all CloudFlare certificates revoked and reissued, 2014. http://blog.cloudflare.com/the-heartbleed-aftermath-all-cloudflare-certificates-revoked-and-reissued.

[32] N. Sullivan. The Results of the CloudFlare Challenge, 2014. http://blog.cloudflare.com/the-results-of-the-cloudflare-challenge.

[33] The GnuTLS Transport Layer Security Library. http://www.gnutls.org.

[34] E. Topalovic, B. Saeta, L.-S. Huang, C. Jackson, and D. Boneh. Toward short-lived certificates. In *Web 2.0 Security & Privacy (W2SP)*, 2012.

[35] S. Yilek, E. Rescorla, H. Shacham, B. Enright, and S. Savage. When private keys are public: Results from the 2008 Debian OpenSSL vulnerability. In *ACM Internet Measurement Conference (IMC)*, 2009.

[36] P. Zheng. Tradeoffs in certificate revocation schemes. In *ACM Computer Communication Review (CCR)*, 2013.

[37] ZMap Vulnerable Hosts. https://zmap.io/heartbleed/vulnerable.html.