In Log We Trust: Revealing Poor Security Practices with Certificate Transparency Logs and Internet Measurements

Oliver Gasser[1], Benjamin Hof[1], Max Helm[1],
Maciej Korczynski[2], Ralph Holz[3], and Georg Carle[1]

[1] Technical University of Munich
[2] Grenoble Alps University & Delft University of Technology
[3] The University of Sydney

Abstract. In recent years, multiple security incidents involving Certificate Authority (CA) misconduct demonstrated the need for strengthened certificate issuance processes. Certificate Transparency (CT) logs make the issuance publicly traceable and auditable.
In this paper, we leverage the information in CT logs to analyze if certificates adhere to the industry's Baseline Requirements. We find 907 k certificates in violation of Baseline Requirements, which we pinpoint to issuing CAs. Using data from active measurements we compare certificate deployment to logged certificates, identify non-HTTPS certificates in logs, evaluate CT-specific HTTP headers, and augment IP address hitlists using data from CT logs. Moreover, we conduct passive and active measurements to carry out a first analysis of CT's gossiping and pollination approaches, finding low deployment. We encourage the reproducibility of network measurement research by publishing data from active scans, measurement programs, and analysis tools.

Keywords: TLS, Certificate Transparency, Baseline Requirements

1 Introduction

One of the Internet's most important protocols, Transport Layer Security (TLS), relies critically on server certificates being issued with diligence by the Web's trust anchors, the Certificate Authorities. It had long been suspected that this degree of trust may be misplaced [13], but from late 2008 on a string of security incidents relating to poor certification practices [31] culminated in the compromise of the DigiNotar Certificate Authority [16]. Being one of the affected parties and a major player on the WWW, Google began work in the IETF on Certificate Transparency (CT) as a response. While this technology is not designed to prevent actual attacks from happening, it can reduce the time to detection drastically. CT essentially turns the Web PKI inside out: a number of independent and neutral logs keep track of issued certificates. This enabled an unprecedented degree of transparency: both certificate misissuance and CA malpractice can now be detected by site operators and third parties. In the years since DigiNotar,

Certificate Transparency has won widespread support. Browser vendors take incidents and malpractice seriously: a number of CAs have been called out for poor practices [37, 27], and the CA PROCERT has been removed from Mozilla's products due to violations of the industry's Baseline Requirements [24]. In this paper, we carry out a thorough analysis of certificates stored in CT and assess CA compliance with the Baseline Requirements.

Main contributions: We perform Internet-wide scans to 196 M hosts, download more than 600 M entries from CT logs, and conduct passive measurements at two different vantage points. Analyzing these data sources, we find 907 k non-expired certificates in violation of Baseline Requirements, and show the proportion of offending certificates is decreasing over time. We quantify the number of domains affected by the impending Symantec distrust. To the best of our knowledge, we conduct the first analysis of non-HTTPS certificates in CT logs and find low rates of log inclusion. We make analysis data, source code, and IP addresses generated from CT log data publicly available to encourage reproducibility in research.

Outline: In Section 2 we give technical background on TLS and CT. Section 3 lays out related work in the Certificate Transparency and certificate analysis fields. In Section 4 we describe our methodology. In the following two sections we analyze the acquired data: Section 5 highlights adherence of CT log certificates to the CA/Brower Forum Baseline Requirements. In Section 6 we compare certificates from CT logs to those from active scans. Section 7 lays out results from investigating CT gossip approaches. We conclude our paper in Section 8.

2 Background

In this section we provide information on protocols relevant for this study.

In order to provide an industry standard for the behavior of CAs in the context of HTTPS, the CA/Browser Forum continuously negotiates technical policies for CA operations. Supplementing specifications such as RFC 5280, it publishes the Baseline Requirements (BRs) [5]. The Baseline Requirements specify important properties for Internet security, for example which algorithms used in certificates are considered secure or what the maximum life time of a certificate may be.

Certificate Revocation Lists (CRLs, see RFC 5280) provide a mechanism to withdraw trust from misissued certificates, e.g., in case of a key compromise.

Repeated misissuances of certificates have led to substantial scrutiny of CAs [9]. Certificate Transparency (CT, see RFC 6962) is a measure to monitor CA behavior. In CT, certificates are submitted into untrusted, public, append-only logs. The primary goal of CT is to allow site operators to observe which certificates were issued for their DNS names. To do this, they inspect the logs, retrieving and examining all certificates included in them. A secondary goal is improving compliance of CAs by easing discovery of misissuances.

On submission of a certificate, the log returns a signed inclusion promise called Signed Certificate Timestamp (SCT). Sites attach the SCT when presenting their certificate, notifying the browser of their participation in CT. Logs regularly

produce signed commitments to a fixed entry list (Signed Tree Heads, STHs). A certificate is considered included in a log when it is covered by an STH.

Today, the Chrome browser requires CT for "Extended Validation" certificates. Starting April 2018, CT will be required by Chrome for all newly issued certificates [34]. Public logs for this purpose are operated by Google and some certificate authorities.

A possible attack by a CT log server is presenting different views to different parties, also called equivocation. This can be addressed with gossip protocols, where participants inform others about the log view presented to them. One such proposal for CT exchanges SCTs and STHs via defined API endpoints on HTTPS servers [29]. The Chrome browser implements an alternative model, where STHs are transferred to the browser via the internal component updater [35]. Inclusion proofs are requested via a custom DNS-based protocol [22].

3 Related Work

The analysis of TLS certificates has become increasingly important, in particular with HTTPS becoming a *de facto* protocol for the Web and many of its APIs [15]. A number of analyses have been carried out, most commonly based on active scans and sometimes passive traffic observation. Our methodology relies to a large degree on a new, different data source, namely CT logs.

Several published works also exploit CT logs, albeit with different research questions. *Amann et al.* [4] examine the use of Certificate Transparency in the context of general improvements to the TLS ecosystems since 2011, a year with a number of major CA incidents. The authors' focus is on the deployment and practical use of these improvements; they do not investigate the properties of logged certificates. *Aertsen et al.* [3] use CT logs to analyze the rise of the Let's Encrypt CA and the resulting more wide-spread use of encryption that enables smaller websites and hosting providers to acquire free certificates. *Gustafsson et al.* [19] use CT logs in combination with passive traffic monitoring to analyze the basic properties of logs and certificates, such as signature algorithm and key lengths of certificates. They do not investigate violations of issuance standards. *VanderSloot et al.* [42] combine CT logs with seven other certificate collection techniques to obtain a picture of the overall HTTPS ecosystem and how different data sources help to make it accurate. They conclude that no collection method covers all certificates. However, they observe that CT logs in combination with active scans cover 98.5 % of their certificates. In our work we make use of this finding to also leverage CT logs and active scans.

A number of earlier publications investigates properties of certificates and TLS deployment. *Holz et al.* [20] provides the first large-scale, long-term analysis of this kind; *Durumeric et al.* [11] later extends this approach to the entire IPv4 space. The publications focus on basic properties of TLS certificates such as weak encryption keys, invalid path length constraints, invalid validity periods, and revoked certificates and sibling CA certificates. *Chung et al.* [8] use TLS scans to analyze certificates without a valid root. They show that invalid certificates make

Table 1: Overview of conducted measurements and used data sources.

Data source	Time period	# Entries	Size
CT log downloads	until Oct. 9, 2017	600 M entries	732 GB
Active HTTPS scans			
IPv4	Oct. 3–8, 2017	196.3 M hosts[1]	259.1 GB
IPv6	Oct. 1, 2017	8.8 M hosts[1]	73.0 GB
CRL downloads	Oct. 11, 2017	25.3 M entries	1.9 GB
Passive CT over DNS			
MWN UDP/53	Sep. 20–27, 2017	2.3 G pkts	10.5 TB
DNSDB TXT #1	Jul. 2016	36.4 M RRs	6.0 GB
DNSDB TXT #2	Sep. 20, 2017	2.4 M RRs	429.8 MB

1: unique IP–domain tuples, e.g., (216.58.207.142,google.com).

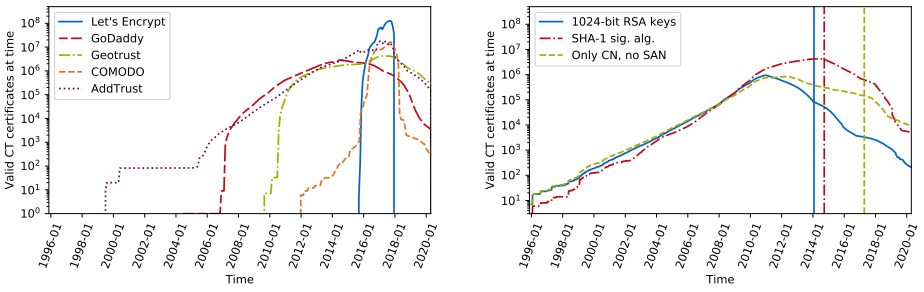
up the majority of collected certificates. A large-scale study of HTTPS-induced browser errors was carried out by *Acer et al.* [1].

4 Methodology

In this section we present our methodology for conducting active and passive measurements. We use various different sources to get a large view of the certificate universe: We download certificates from CT logs, obtain certificates from active scans, retrieve CRLs, and conduct active and passive measurements to analyze CT gossiping deployment. Table 1 gives an overview of these sources, detailing the time of data collection, the number of entries, and the size of the acquired data. We also detail ethical and reproducibility considerations.

CT Log downloads. We extend Google's CT tool to incrementally download certificates and their certificate chains from 30 CT logs. We publish our extended CT tool on GitHub [39]. In total we download 600 M log entries, resulting in 216.8 M unique certificates and 7.8 M unique certificates in chains.

Active HTTPS measurements. To compare the certificates seen in CT logs to the actual HTTPS deployment we conduct active measurements over IPv4 and IPv6. First, we collect a total of 1.2 G domains from three different sources: TUM's hitlist [18], domains contained in CN and SAN of downloaded CT log certificates, and Farsight's DNSDB [14]. Second, we filter auto-generated disposable domains [6] from the DNSDB data by removing subdomains such as `netflixdnstest1.com` and domains with less than 100 queries within a month as indicated by DNSDB. Third, we resolve the remaining domains to A and AAAA records. Fourth, we conduct port scans on TCP/443 using ZMap [12] for IPv4, and our IPv6-enabled version [41] for IPv6. Fifth, we use our highly parallelized Goscanner [40] to establish TLS connections to 191.4 M and 8.8 M IP address–domain name tuples for IPv4 and IPv6, respectively. To obtain the correct certificate we send the domain name in the SNI extension. Upon successful connection establishment we send HTTP requests to retrieve the server's HTTP headers and check for the presence of gossiping and pollination endpoints [29].



(a) Certificates per CA. (b) Certificates in violation of BRs.

Fig. 1: Non-expired certificates in CT logs, by issuing CA and conformance with three BRs (vertical line is Chrome enforcement date). Y-axis is log-scaled.

CRL downloads. In order to determine the revocation status of certificates, we extract CRL URLs from certificates of active scans and CT logs. We then download these CRL files as well as Mozilla's OneCRL [28]. In total we extract 25.3 M entries from CRLs. We do not check OCSP as it is disabled in Chrome and previous work shows limited support [23].

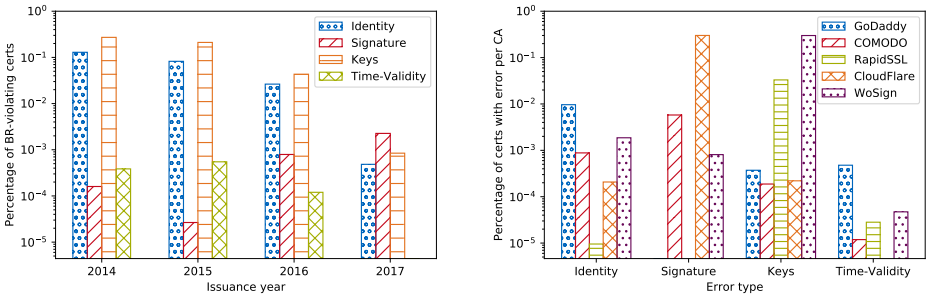
Passive DNS measurements. To analyze the use of Google's CT over DNS approach [22], we conduct passive measurements. We evaluate one week of DNS traffic at the Internet uplink of the Munich Scientific Network. Additionally, we use Farsight's DNSDB data [14] to further improve our client coverage.

Ethical considerations. We follow an internal multi-party approval process before any measurement activities are carried out. This process incorporates the proposals of Partridge and Allman [30] as well as Dittrich *et al.* [10]. We assess whether our measurements can induce harm on individuals in different stakeholder groups. As we limit our query rate and use conforming HTTP requests, it is unlikely for our measurements to cause problems on scanned systems. Using the REST API provided by CT logs, we perform incremental downloads to reduce the impact on target systems. We follow best scanning practices such as maintaining a blacklist and using dedicated servers with informing rDNS names, web sites, and abuse contacts. We limit our passive measurements to DNS TXT records. The conclusion of this process is that it is ethical to conduct the measurements, but that we will only share data from our active measurements and not release passive data to protect the privacy of involved parties.

Reproducible research. To encourage reproducible research in network measurements [2, 33], we publish source code and data in the long-term availability archive of the TUM University Library: https://mediatum.ub.tum.de/1422427

5 Baseline Requirements

In this section we analyze the certificates found in CT logs, with a particular focus on their compliance with the Baseline Requirements. Figure 1a shows the result



(a) Violation category over issuance.

(b) Violation category per CA.

Fig. 2: Proportion of certificates in violation of BR. Y-axis is log-scaled.

of a quantitative analysis of non-expired certificates of the top 5 CAs over time. As is to be expected, the number of current, non-expired certificates peaks for most CAs around our cut-off date of October 9, 2017. One exception is GoDaddy, whose number of issued, non-expired certificates has been decreasing since 2014. We see that the vast majority of certificates in logs are issued by Let's Encrypt (LE), which saw exponential growth after the service became publicly available in 2016. Furthermore, due to the 3 month validity period of LE certificates, a sharp decline of certificates can be seen at the beginning of 2018. Due to longer validity periods, this decline is less pronounced for other CAs.

To evaluate the conformance of certificates to BRs, we run the cablint tool [38] on all non-expired certificates found in CT logs. We find 907 k certificates (1.3 %) in violation of BRs. Three major security relevant changes in the last years are shown in Figure 1b, with vertical bars denoting deprecation steps by the Chrome browser. We observe that the prohibition of practices such as short keys is followed by a substantial reduction in the number of affected certificates. It takes years, however, until all old non-compliant certificates are expired.

Next, we look at violations of requirements or recommendations in the current BRs. We categorize these violations as pertaining to the *identity* (e.g., SAN or CN), *signature* (e.g., hash algorithm), *key* (e.g., key usage or size), or *validity time*. Grouping certificates by year of issuance, Figure 2a shows the proportion of certificates exhibiting errors in these categories. This allows us to see the proportion of problematic certificates independent of the issuance rate. Generally, the proportion of certificates with errors is declining over time, with identity and key issues being predominant. In 2017, signature related issues become the prevalent cause of errors.

Attributing these violations to specific CAs, we select the 5 CAs with the highest number of infringing certificates. We show the number of violating certificates in the different categories per CA relative to their total issued certificates in Figure 2b. The most significant infractions are SHA1 signatures by CloudFlare and use of non-critical key usage extensions by WoSign. Upon closer investigation we find that most certificates with BR violations are signed by revoked intermediate certificates. We use our measurement results to improve issuance practices

Table 2: Comparison of certificates found in CT logs and active scans.

Cert source	Total	Not revoked	Not expired	Not self-signed	Browser-valid	BR-valid
CT logs	216.8 M	216.6 M	70.2 M	216.8 M	70.2 M	206.3 M
Active scans	128.1 M	127.5 M	118.8 M	109.4 M	74.8 M	115.4 M

by notifying affected CAs. Furthermore, we note that Let's Encrypt has never committed any BR violations, while issuing the most certificates. Their service therefore improves Internet security not only by democratizing encryption [3], but by doing so in exemplary accordance with best practices.

6 Comparing CT Log Data to Active Scans

In this section we evaluate the differences between certificates in CT logs and those obtained from active scans dating back until 2009. Additionally, we take a first look at the deployment of CT-specific HTTP headers and determine the value of CT logs to create IP address hitlists.

6.1 Certificate Deployment and Validity

In our active scans we collect 316.3 M certificates (32.8 M unique) from 128.3 M successful handshakes with IPv4 hosts and 4.2 M IPv6 hosts. When the same certificate is presented for a name under all its IP addresses, within and across IP versions, we call the domain *consistent*. The vast majority of domains (e.g., 99 % for IPv6) delivers consistent certificate chains. We investigate inconsistent domains and find that these are mostly due to TLS services offered by Content Distribution Networks (CDNs): 86.9 % of IPv6 inconsistencies can be attributed to CloudFlare, 5.4 % to Akamai. Inconsistent chains use the same certificate key and *Common Name* in about 80 % of the cases. *Subject Alternative Name* entries, however, are deviating to a large extent. We conclude that inconsistent certificate chains are mostly due to CDNs dynamically adding client domains to certificates. In the following we limit our analysis to the 128.1 M consistent domains in order to make quantitative statements more intuitively understandable.

We analyze the overlap of certificates in CT logs and certificates obtained from active scans and find that 109.8 M (85.7 %) certificates from active scans are logged in CT. This high percentage is an encouraging milestone towards the goal of logging all deployed certificates. Starting April 2018, the Chrome browser will only accept newly issued certificates which are logged [34].

In Table 2 we distinguish certificates by revocation, expiration, self-signed, browser-valid status, as well as conformance with the Baseline Requirements.

For CT log and active scan certificates, we find low numbers of certificates revoked through embedded CRLs or OneCRL [28].

More than 92 % of certificates found in active scans are not expired. In CT logs, however, more than two thirds of certificates are expired. This is to be

expected, since CT logs explicitly keep expired certificates. This feature allows to easily evaluate trends in the certificate ecosystem over time.

The picture changes when evaluating self-signed certificates: CT logs only accept certificates valid under root stores and therefore do not contain self-signed server certificates. In active scans we find 14.6 % self-signed certificates, which is a decrease compared to previous studies [8, 11]. This could be an indicator of Let's Encrypt's democratizing impact [3], where the lower end of the market moves from self-signed to free CA-signed certificates.

Next, we analyze whether certificates are accepted by web browsers. These are a subset of certificates which are neither revoked nor expired nor self-signed. Additional conditions (e.g., matching domain, correct chain to root cert) must be met as well. Since CT logs only accept root store-anchored certificates, all valid CT log certificates are accepted by browsers. However, only 63 % of not expired certificates from active scans are browser-valid. Therefore a non-negligible number of certificates found in the wild is resulting in security warnings to users.

Moreover, we compare BR violations of certificates found in CT logs and found using active scans. 95.2 % of logged certificates are valid according to the BRs, compared to 90.1 % of deployed certificates. This finding underlines the importance of logging all certificates in order to make violations more easily traceable and CAs more accountable.

Furthermore, we assess the impact of the impending distrust of Symantec root certificates [26]. We find 4.2 M domains where one of the Symantec root certificates is used. Limiting our analysis to specific certificate validity periods allows us to quantify the impact more precisely: 1.9 M domains will not be trusted anymore in May 2018, whereas 777.7 k domains will be affected by the complete removal of Symantec root certificates in October 2018. These findings show that many domains have not yet switched to other CAs and stress the importance of a smooth transition to the new Symantec CA owner DigiCert.

6.2 Legacy and Non-HTTPS Certificates in CT Logs

We use our data sets from previous work [20, 21] to check how many certificates from scans dating back as far as November 2009 have been included in CT logs. Table 3 summarizes the results. A surprising number of older certificates are indeed contained in CT logs. More than 21% of certificates used on HTTPS-secured domains on the Alexa Top 1M list from 2009 are in CT logs. Their median expiry time is May 2011; this is well before CT was even deployed. It is known that Google scanned the Internet relatively regularly to bootstrap and fill CT logs. Of certificates retrieved in 2011, more than half are in CT logs, even though their median expiry time is the first half of 2012—CT was not even standardized then. This shows that CT logs were filled early with certificates that would already be of little use once actual CT deployment would start.

The scans conducted in 2015 [21] also considered email, messaging, and file transfer protocols. These scans provide us with insights about the logging of non-HTTPS certificates. We find a clear trend: certificates found solely in a non-HTTPS scan are generally not included in CT logs, only 3.5 % or less. Certificates

Table 3: Presence of certificates from previous work in CT logs. For the IPv4-wide scans, we group protocol-wise, combining scans via STARTTLS and direct TLS into one group. The table is to be read from left to right, i.e., 'Logged' means 'not already in HTTP scan and logged in CT'. Median expiry then refers to the latter group.

Scan	# Certs	Not in HTTP	Logged	Median expiry
Alexa, HTTPS, 2009	248.3k	n/a	53.3k (21.5%)	2011-05-05
Alexa, HTTPS, 2011	222.1k	n/a	126.9k (57.1%)	2012-02-29
IPv4, 2015	11.3M	n/a	3.2M (27.9%)	2016-03-18
HTTPS	8.8M	n/a	3.1M (35.1%)	2016-03-18
SMTPS	1.7M	1.2M (69.4%)	57.8k (3.5%)	2016-04-25
IMAPS	1.3M	893.3k (71.2%)	41.2 (3.3%)	2016-04-27
POP3S	1.1M	814.5k (72.3%)	32.1k (2.8%)	2016-04-09
FTPS	753.2k	597.0k (79.3%)	21.3k (2.8%)	2016-02-01
XMPPS	67.2k	51.6k (76.8%)	1.3k (2.0%)	2016-05-14
IRCS	7.4k	6.0k (81.2%)	181 (2.5%)	2016-01-06

that we found to be used for both HTTPS and non-HTTPS services are logged a bit more often: between 9.1 % (SMTPS) and 8.5 % (XMPPS and IRCS) fall into this category.

6.3 CT-Specific HTTP Headers

Similarly to enforcing HTTPS-only connections using the HTTP Strict Transport Security (HSTS) header (see RFC 6797), web servers can require the presence of certificates in CT logs. Requiring the presence in logs allows to detect man-in-the-middle attacks where the original server certificate is replaced by an attacker. We analyze the deployment of the unofficial RequireCT [32] and the draft RFC Expect-CT [36] headers.

We find eight domains sending HSTS headers with a RequireCT directive and 7.3 k domains with Expect-CT headers. In the following, we investigate the Expect-CT deployment. This header consists of a mandatory *max-age* field and optional *enforce* and *report-uri* fields. We find 12.1 % of domains to omit the mandatory *max-age* directive. The majority of domains sets the *max-age* to zero, effectively disabling the Expect-CT mechanism. Only 29.9 % of domains *enforce* Expect-CT, the majority makes only use of the reporting feature. With 608 domains, less than 10 % enforce Expect-CT with a duration of one day or more.

We check whether domains which send an Expect-CT header have in fact logged their certificate in CT. The majority of certificates can be found in CT logs. However, 83 Expect-CT domains (1.2 %) do not send certificates which are logged. 48 of these enforce Expect-CT with a *max-age* greater than zero. These domains do not comply with the Expect-CT specification. We find a lower misconfiguration percentage in Expect-CT compared to the more established yet complex public key pinning via HPKP headers [4].

6.4 CT Logs as a Source for IP Address Hitlists

CT logs contain not only valuable information about certificates, but are also an additional source of domain names. We analyze the value of domain names extracted from CN and SAN of logged certificates by comparing them to our publicly available hitlist [18]. TUM's hitlist provides IP addresses based on domains from zonefiles, Alexa Top 1M, Cisco Umbrella, CAIDA, and Rapid7.

The CT log data adds 82.2 M domains, 5.4 M IPv4, and 489 k IPv6 addresses to the hitlist. This corresponds to respective increases of 50.5 %, 56.2 %, and 69.6 %. Especially the large increase of IPv6 addresses can aid future measurement studies. We make the hitlist enhanced with CT domain data freely available [17].

7 Gossiping and Inclusion Proofs

CT offers gossiping protocols to detect equivocation attacks, where a log presents different views to different parties. Gossiping allows clients to exchange their log view with each other. Clients can also request inclusion proofs from the log, demonstrating that a specific certificate was indeed incorporated by the log. We conduct active and passive measurements to evaluate if these techniques are used.

As part of our active scans, we send HTTP requests to responding domains in order to evaluate the deployment of CT gossiping endpoints among HTTPS websites. These requests are targeted at specific URL paths used in CT gossiping [29]. Additionally, we send one request to a non-existent path that serves as the baseline of how web servers answer requests for non-existent paths.

In the course of these measurements, we receive answers from 109.2 M domains and inspect the HTTP return codes. We remove hosts that answer with 2xx or 3xx to the non-existent baseline path, send the same answer for CT paths as the baseline request, or answer with 4xx to the CT paths. After this filtering 16.8 k (0.015 %) domains remain. This is an upper bound of domains supporting HTTP-based CT gossiping, as web servers might be configured in a way which triggers different behavior for CT and the baseline path. To lower this upper bound, more complex measurements would need to be performed. These low numbers, however, suggest that HTTP-based gossiping is not widespread.

The gossip requests generated a magnitude more abuse notifications compared to other scans. This should be considered in the protocol specification, e.g., by using an HTTP header as a discovery mechanism less prone to undue excitement. Alternatively, browsers could gradually acclimate operators to this new reality.

In addition to active HTTPS scans, we conduct passive DNS measurements as described in Section 4. Since HTTPS URL paths are encrypted in TLS and therefore not visible, we instead evaluate the deployment of Google's proposal to fetch inclusion proofs over DNS [22]. Even though the CT over DNS proposal is implemented in Google's Chrome browser [7], we could not find any TXT record matching the document specification in our passive data. This was confirmed by Google, who said they never activated the protocol due to privacy concerns [25].

We conclude that protection against split-view attacks by logs which is an architectural necessity in CT has next to no deployment in the wild.

8 Conclusion

In this study we investigated the Baseline Requirements adherence of certificates found in CT logs and through active scans. We mapped these violations to issuing CAs and inform them of our findings. Furthermore, we compared the results from CT logs and active scans, finding that logged certificates exhibit less violations. We note that the log inclusion rate of non-HTTPS certificates is significantly lower. Additionally, we observed that CT gossiping, although required in the security model of CT, does currently not have any substantial deployment.

Acknowledgments: We thank Emily Stark from Google for the valuable insights into Chrome's current state of CT over DNS. The authors thank the contributors of data to Farsight Security's DNSDB. We thank the anonymous reviewers and our shepherd Steve Uhlig for their valuable feedback. This work was partially funded by the German Federal Ministry of Education and Research under project X-Check, grant 16KIS0530, and project DecADe, grant 16KIS0538.

References

[1] M. E. Acer et al. "Where the Wild Warnings Are: Root Causes of Chrome HTTPS Certificate Errors". In: *CCS '17*.

[2] ACM. *Artifact Review and Badging*. 2016. URL: https://www.acm.org/publications/policies/artifact-review-badging.

[3] M. Aertsen et al. "No domain left behind: is Let's Encrypt democratizing encryption?" In: *ANRW '17*.

[4] J. Amann et al. "Mission Accomplished? HTTPS Security after DigiNotar". In: *IMC '17*.

[5] CA/Browser Forum. *Baseline Requirements for the Issuance and Management of Publicly-Trusted Certificates*. Version 1.5.0. Sept. 1, 2017.

[6] Y. Chen et al. "DNS noise: Measuring the Pervasiveness of Disposable Domains in Modern DNS Traffic". In: *DSN '14*.

[7] Chromium authors. *CT over DNS implementation in Chromium*. URL: https://cs.chromium.org/chromium/src/components/certificate_transparency/log_dns_client.cc?type=cs&sq=package:chromium.

[8] T. Chung et al. "Measuring and applying invalid SSL certificates: The Silent Majority". In: *IMC '16*.

[9] J. Clark and P. van Oorschot. "SoK: SSL and HTTPS: Revisiting Past Challenges and Evaluating Certificate Trust Model Enhancements". In: IEEE S&P '13.

[10] D. Dittrich et al. "The Menlo Report: Ethical Principles Guiding Information and Communication Technology Research". In: *US DHS* (2012).

[11] Z. Durumeric et al. "Analysis of the HTTPS Certificate Ecosystem". In: IMC '13.

[12] Z. Durumeric et al. "ZMap: Fast Internet-wide Scanning and Its Security Applications". In: *USENIX Security '13*.

[13] C. Ellison and B. Schneier. "Ten Risks of PKI: what You're not Being Told about Public Key Infrastructure". In: *Computer Security Journal* (2000).

[14] Farsight Security. *DNSDB*. URL: https://www.dnsdb.info/.

[15] A. P. Felt et al. "Measuring HTTPS Adoption on the Web". In: *USENIX Security '17*.

[16] Fox-IT. *Black Tulip. Report of the investigation into the DigiNotar Certificate Authority breach*. Aug. 2012.

[17] O. Gasser et al. *IPv6 Hitlist Collection*. URL: https://www.net.in.tum.de/projects/gino/ipv6-hitlist.html.

[18] O. Gasser et al. "Scanning the IPv6 Internet: Towards a Comprehensive Hitlist". In: *TMA '16*.

[19] J. Gustafsson et al. "A First Look at the CT Landscape: Certificate Transparency Logs in Practice". In: *PAM '17*.

[20] R. Holz et al. "The SSL landscape—a thorough analysis of the X.509 PKI using active and passive measurements". In: *IMC '11*.

[21] R. Holz et al. "TLS in the wild - An Internet-wide analysis of TLS-based protocols for electronic communication". In: *NDSS '16*.

[22] B. Laurie et al. *Certificate Transparency RFCs on GitHub*. 2017. URL: https://github.com/google/certificate-transparency-rfcs.

[23] Y. Liu et al. "An End-to-End Measurement of Certificate Revocation in the Web's PKI". In: *IMC'15*.

[24] G. Markham. *Mailing List: Mozilla dev.sec.policy: PROCERT decision*.

[25] E. Messeri. *Mailing List: IETF Trans: Privacy analysis of the DNS-based protocol for obtaining inclusion proof*.

[26] Mozilla. *Mailing List: Mozilla dev.sec.policy: Mozilla's Plan for Symantec Roots*.

[27] Mozilla. *Revoking Trust in Two TurkTrust Certificates*. URL: https://blog.mozilla.org/security/2013/01/03/revoking-trust-in-two-turktrust-certficates/.

[28] *Mozilla OneCRL*. Oct. 2017.

[29] L. Nordberg et al. *Gossiping in CT*. Internet-Draft draft-ietf-trans-gossip-04.

[30] C. Partridge and M. Allman. "Ethical Considerations in Network Measurement Papers". In: *Communications of the ACM* (2016).

[31] I. Ristić. *SSL/TLS and PKI History*. URL: https://www.feistyduck.com/ssl-tls-and-pki-history/.

[32] T. Ritter. *An Experimental "RequireCT" Directive for HSTS*. Feb. 2015. URL: https://ritter.vg/blog-require_certificate_transparency.html.

[33] Q. Scheitle et al. "Towards an Ecosystem for Reproducible Research in Computer Networking". In: *SIGCOMM Reproducibility '17*.

[34] R. Sleevi. *Certificate Transparency in Chrome - Change to Enforcement Date–Google Groups*. Apr. 21, 2017. URL: https://groups.google.com/a/chromium.org/forum/#!msg/ct-policy/sz_3W_xKBNY/6jq2ghJXBAAJ.

[35] R. Sleevi and E. Messeri. *Certificate Transparency in Chrome: Monitoring CT Logs Consistency*. May 1, 2015. URL: https://docs.google.com/document/d/1FP5J5Sfsg0OR9P4YT0q1dM02iavhi8ix1mZlZe_z-ls.

[36] E. Stark. *Expect-CT Extension for HTTP*. Internet-Draft draft-ietf-httpbis-expect-ct-02.

[37] Symantec. *Update on Test Certificate Incident*. 2016. URL: https://www.symantec.com/page.jsp?id=test-certs-update.

[38] TUM. *cablint on GitHub*. URL: https://github.com/tumi8/certlint.

[39] TUM. *CT Go Tool on GitHub*. URL: https://github.com/google/certificate-transparency-go.

[40] TUM. *goscanner on GitHub*. URL: https://github.com/tumi8/goscanner.

[41] TUM. *ZMapv6 on GitHub*. URL: https://github.com/tumi8/zmap.

[42] B. VanderSloot et al. "Towards a Complete View of the Certificate Ecosystem". In: *IMC'16*.