Characterizing the Root Landscape of Certificate Transparency Logs

Nikita Korzhitskii* Niklas Carlsson†
Linköping University, Sweden
Email: *nikita.korzhitskii@liu.se, †niklas.carlsson@liu.se

***Abstract*—Internet security and privacy stand on the trustworthiness of public certificates signed by Certificate Authorities (CAs). However, software products do not trust the same CAs and therefore maintain different root stores, each typically containing hundreds of trusted roots capable of issuing "trusted" certificates for any domain. Incidents with misissued certificates motivated Google to implement and enforce Certificate Transparency (CT). CT logs archive certificates in a public, auditable and append-only manner. The adoption of CT changed the trust landscape. As a part of this change, CT logs started to maintain their own root lists and log certificates that chain back to one of the trusted roots. In this paper, we present a first characterization of this emerging CT root store landscape, as well as the tool that we developed for data collection, visualization, and analysis of the root stores. As part of our characterization, we compare the logs' root stores and quantify their changes with respect to both each other and the root stores of major software vendors, look at evolving vendor CT policies, and show that root store mismanagement may be linked to log misbehavior. Finally, we present and discuss the results of a survey that we have sent to the log operators participating in Apple's and Google's CT log programs.**

I. Introduction

The end-to-end security and privacy provided by HTTPS heavily depends on the trustworthiness of X.509 certificates and whether these certificates were signed by a Certificate Authority (CA) trusted by the client's software. For example, most browsers establish a connection using a public key from a certificate only if (i) it was directly signed by a CA included in their root store, or (ii) if it was signed by a sequence of intermediate CAs that chains back to one of these trusted roots. However, not all CAs are equally trustworthy and software vendors of browsers have decided to use different root stores. Furthermore, successful attacks, misconfigurations and other incidents have significantly reduced the trust in CAs [22].

Certificate Transparency (CT) [21] was recently introduced as a response to waning CA trust. It provides an additional measure to improve the general trust in certificates, is already successfully deployed, and today plays a central and increasingly important role in the certificate ecosystem. With CT, certificates are logged in public, auditable, append-only CT logs. This helps domain owners to discover malicious or misissued certificates soon after they are issued. Certificates can be submitted to logs by any party. Upon submission logs issue Signed Certificate Timestamps (SCTs), which provide a cryptographic promise that the log will publish the submitted certificate chain within the log's Maximum Merge Delay (MMD). Servers can then deliver SCTs to their clients (e.g., browsers) to prove that a certificate has been logged.

ISBN 978-3-903176-28-7 © 2020 IFIP

Both Google and Apple have implemented their own CT policies. For example, since 2015, as part of Google Chrome's validation process, all newly issued Extended Validation (EV) certificates are required to be logged in (at least) two Chrome-trusted CT logs (one Google-operated log and one non-Google-operated log [14]), and since the release of Chrome v.68 in Jul. 2018, Chrome requires *all* certificates issued after Apr. 2018 to be logged [28]. Similarly, since Oct. 15, 2018, Apple requires all newly issued certificates to be included in several logs [6]. Neither Mozilla or Microsoft have a public CT policy, but some Microsoft products have optional CT support [25], [32].

To comply with the above policies, CAs (and sometimes domain owners) are actively submitting their certificates to logs upon issuance. Over recent years, the CT logs are therefore quickly growing in size. For example, between January 2019 and January 2020, the number of entries in the CT logs tracked by Google increased from ≈3B entries to more than 7.5B entries [16].

By analogy with vendor root stores, there exist many CT logs and log operators configure their own lists of "trusted" or "acceptable" [21] root CAs. Since logs only accept certificates chaining back to these roots, the roots determine which certificates can be logged (and indirectly also which certificates can be trusted by Chrome and Apple products). Some logs also introduce additional submission requirements (e.g., expiration constraints). However, in general, it is important that certificates issued by trusted CAs can be easily submitted to multiple trusted logs.

In this paper, we present a novel characterization of this emerging root landscape and the tool that we developed for data collection, visualization, and analysis of the logs' root stores. First, we provide an overview of the active CT logs, their root stores, describe the changing root store landscape and relate our observations to current CT policies (by Google and Apple) and the major vendor root stores (by Apple, Microsoft, and Mozilla). We observed that root lists are diverse, that the root stores of most logs are increasing in size, becoming more similar, but also that most logs' root stores do not cover a significant fraction of the root stores of major software vendors, and that CT relies heavily on the

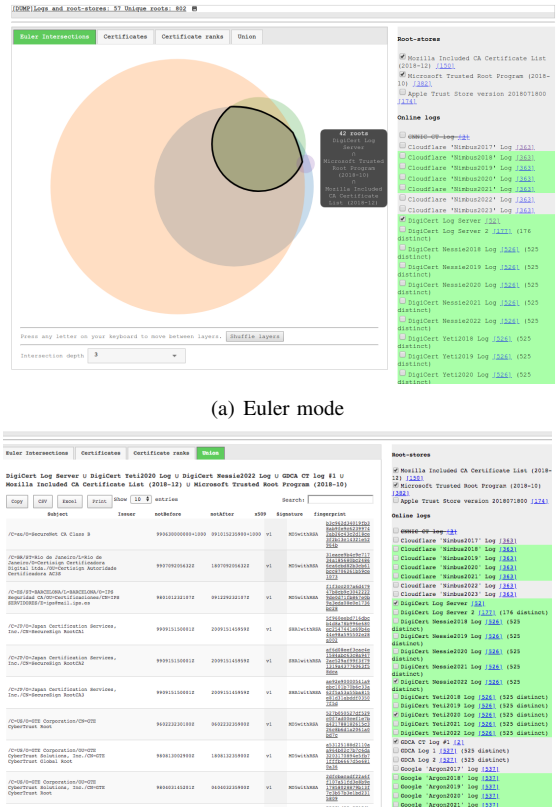
logs operated by only five log operators (Cloudflare, DigiCert, Google, Sectigo, and Let's Encrypt), where the logs of the fifth operator only became qualified by Google on Oct. 7, 2019. This raises questions regarding whether the CT infrastructure is sufficiently redundant and reliable. Second, we identify and highlight instances of potential log mismanagement, including some instances that were followed by log misbehavior events resulting in logs being distrusted. Third, we highlight a number of notable roots and specific log behaviors that demonstrate the diversity in the ways the logs are used. The discussion is followed by the results of our survey of the five log operators with Apple- and Google-trusted CT logs. Finally, along with this paper, we publish our interactive, online, open-source tool and a longitudinal dataset to allow others to further investigate the root stores and their relations.

The paper is organized as follows. Section II describes how Chrome and Apple decide which logs to trust. Section III presents the tool and dataset. Section IV characterizes logs' root stores, their relations to each other and to the stores of major software vendors, as well as how the landscape is changing. The following section presents our survey results (Section V-A), highlights log mismanagement and misbehavior (Section V-B), as well as notable roots and use cases (Section V-C). Finally, we present related work (Section VI) and conclusions (Section VII).

II. Trusted vs Non-trusted Logs

Both Google and Apple maintain their own CT policies and lists of trusted CT logs, with candidate logs only becoming trusted after demonstrating policy compliance over a test period. To become Chrome-trusted (or "qualified"), a log has to comply with the CT standard [21], properly incorporate accepted certificate chains within the MMD, and satisfy availability constraints [14]. Moreover, the root store of every log must include Google's *MMD Root*, which Google's monitoring servers use to perform test submissions. Logs are continually monitored, and Google disqualifies logs that violate policy requirements. A qualified log must accept all certificates that chain back to the log's root store, however, it is allowed to reject submissions on the basis of validity, revocation, and/or expiration status. Finally, Chrome-trusted logs have to publish their root lists and post updates in a corresponding thread of Google's CT bugtracker [13]. However, these announcements intertwine with other log-related posts and do not provide a solid, tamper-proof and machine-readable history of changes in the root lists.

Since Apple's CT policy [6] has no public repository, it is more difficult to track changes over time. However, we have observed some updates to the policy, including a new requirement, stating that "a log must accept certificates that are issued by Apple's compliance root CA to monitor the log's compliance with these policies" [5]. Apple also states that "logs must trust all root CA certificates included in Apple's trust store" [5]. As we show later in the paper, logs do not fully adhere to these two requirements. Both log programs adopt a similar log list schema; participating logs are assigned one of the following states: usable, qualified, read-only, retired, pending or rejected.



(a) Euler mode

(b) Listing mode

Fig. 1. *Certificate Transparency Root Explorer* screenshots: (a) interactively selected intersection in Euler Diagram visualization mode, and (b) certificate-listing mode.

III. Root Explorer and the Dataset

To allow readers to dive deeper into our results and the CT landscape, we will share both our tool and the datasets [19].

Tool: We developed *CT Root Explorer* — a web-based open-source interactive tool for data collection and analysis of logs' root stores, including the relations to each other and to major vendor root stores. The tool can be used both on live data and historic data. The tool first retrieves roots directly from available logs and/or import/export SQLite snapshots. The tool allows a user to interactively visualize and analyze selected root stores, explore certificate frequencies, intersections, complements and unions. Information about roots can also be filtered and exported. Figure 1 shows the tool's interface in *Euler diagram* mode, in which a user can interactively select an intersection between a number of logs, and in *Certificate listing* mode, in which the information about a selected set of roots is listed and can be filtered and exported. The tool also has two other modes, including the *Root frequency* mode, which generates an interactive frequency diagram (an annotated version is presented in Figure 6).

Primary dataset: Since Nov. 2, 2018, we have collected hourly snapshots of the root stores of all logs listed as known

TABLE I
COVERAGE OF VENDOR ROOT STORES AND OTHER PROPERTIES OF AVAILABLE CERTIFICATE TRANSPARENCY LOGS DURING OCTOBER 8TH, 2019 AS COMPARED TO DECEMBER 27TH, 2018

Log	Distinct list	Google Log Program	Apple Log Program	Distinct certificates		Duplicates		Merge Delay Monitor Root	DigiNotar Root CA	Apple, %		Microsoft, %		Mozilla, %		CORS headers	Test submission	Expiration constraint	Rejects expired
Cloudflare Cirrus (RPKI log)	1	not listed	not listed	5				-		0.0		0.0		0.0		-	+		
Cloudflare Nimbus20{17-18}	2	rejected	readonly	576	+213			+	+	100	+9.8	97.5	+29.7	100	+16.7	-	–	+	+
Cloudflare Nimbus20{19-21}	2	usable	usable	576	+213			+	+	100	+9.8	97.5	+29.7	100	+16.7	-	+	+	–
Cloudflare Nimbus20{22-23}	2	qualified	usable	576	+213			+	+	100	+9.8	97.5	+29.7	100	+16.7	-	+	+	
DigiCert Log Server	3	usable	usable	56	+4	1	+1	+		27.5	+1.6	15.7	+3.1	32.9	+2.9	-	+	–	–
DigiCert Log Server 2	4	usable	usable	179	+3	1		+		79.2	-3.0	45.8	+3.7	87.9	-2.8	-	±	–	–
DigiCert Nessie2018	5	rejected	rejected	531	+6	2	+1	+		97.2	-2.2	87.1	-4.0	93.3	-2.0	-	–	+	+
DigiCert Yeti2018	5	rejected	usable	531	+6	2	+1	+		97.2	-2.2	87.1	-4.0	93.3	-2.0	-	–	+	+
DigiCert Nessie20{19-22}/Yeti20{19-22}	5	usable	usable	531	+6	2	+1	+		97.2	-2.2	87.1	-4.0	93.3	-2.0	-	+	+	+
DigiCert Nessie2023/Yeti2023	5	qualified	qualified	531		2		+		97.2		87.1		93.3		-	+	+	
Google Argon2022	6	qualified	usable	561	+24			+		99.4	+0.0	97.2	+2.4	100	+0.7		+	+	
Google Daedalus	6	no state	not listed	561	+24			+		99.4	+0.0	97.2	+2.4	100	+0.7		+	+	–
Google Icarus	7	usable	usable	3				+		1.1	+0.0	0.6	+0.1	1.3	+0.0		+	–	–
Google Skydiver	8	usable	usable	559	+24			+		98.3	+0.0	96.6	+2.4	98.7	+0.7		+	–	–
Google Solera2023	9	no state	not listed	225				+		1.1		0.6		1.3			+	+	
Google Submariner	10	no state	not listed	81	+1			+		20.8	-1.6	20.3	+3.0	15.4	+0.1		+	–	–
Google Crucible/Solera20{18-22}/Testtube	9	no state	not listed	225	+31			+		1.1	+0.0	0.6	+0.1	1.3	+0.0		+		
Google Argon2017	6	rejected	usable	561	+24			+		99.4	+0.0	97.2	+2.4	100	+0.7		–	+	+
Google Argon2018/Xenon2018	6	rejected	usable	561	+24			+		99.4	+0.0	97.2	+2.4	100	+0.7		+	+	–
Google Argon20{19-21}/Xenon20{19-22}	6	usable	usable	561	+24			+		99.4	+0.0	97.2	+2.4	100	+0.7		+	+	–
Google Pilot/Rocketeer	6	usable	usable	561	+24			+		99.4	+0.0	97.2	+2.4	100	+0.7		+	–	–
Google Argon2023/Xenon2023	6	qualified	qualified	561				+		99.4		97.2		100			+	+	
LetFLs Encrypt Oak20{19-22}	11	qualified	qualified	412				+		100		99.4		100			+	+	+
Nordu Plausible	12	no state	not listed	444				+		81.5	-4.7	68.3	-6.6	76.5	-2.8	-	+	–	–
Sectigo Dodo	13	no state	not listed	522	+73			+		100		100	+0.3	100	+0.7	-	+	–	–
Sectigo Mammoth/Sabre	14	usable	usable	371	+14			+		100		89.8	+0.8	98.7	-0.6	-	+	–	–
WoTrus	15	not listed	not listed	9				+		1.7		0.0		0.0		-	–		
CNNIC CT log		retired	retired	–													–		
GDCA Log 1/Log 2		rejected	rejected	–		2											–		
GDCA CT log #1/SHECA CT log 2		rejected	not listed	–													–		
Google Aviator		readonly	readonly	–													–		
Venafi Gen2 CT log		readonly	readonly	–		6											–		
Apple Trust Store version 2018071800				174						94.9		46.2		81.2					
Apple Trust Store version 2018121000				178						100		47.4		85.2					
Microsoft Trusted Root Program (2018-10)				382						91.6		96.6		97.3					
Microsoft Trusted Root Program (2019-07)				325						86.5		100		97.3					
Mozilla Included CA Certificate List (2018-12)				150						74.2		44.6		96.6					
Mozilla Included CA Certificate List (2019-10)				149						71.3		44.6		100					

Positive and negative colored values designate change between the initial and final measurements. "+"/"-" in *Test submission* column designate whether the submission was successful. The symbol "±" indicates that only some submissions were successful.

by Google [17], [15] and Apple [5] using *CT Root Explorer* running on Chromium v.71. The default snapshots used in the analysis presented here (and made available with the tool) were collected on Dec. 27, 2018, and Oct. 8, 2019. These snapshots include the root stores of 54 and 57 logs, respectively. We also include corresponding snapshots of the three popular vendor root stores by Apple [3], [4], Microsoft [24] and Mozilla [26]. (Unlike Firefox, Google Chrome relies on the root store of the underlying operating system, but reserves the right to distrust selected CAs [12].) Table I lists the collected logs and root stores. Here, some logs with identical properties are grouped by a year range (e.g., Argon20{18-20}), where the year typically reflects an additional expiration criteria on the certificates to be logged. We also include *Cloudflare Cirrus* and *WoTrus* ("not listed"), which are not listed as *known* by Apple or Google. For each log, we specify their current status in Google's and Apple's log programs. Here, we distinguish between trusted logs (green) that are (i) usable, and (ii) qualified (i.e., accepted, but not yet used). Non-trusted logs include logs that are (i) rejected, (ii) not listed, or have (iii) no state. At the boundary of this class (yellow) we include (iv) retired (previously usable/qualified) and (v) read-only (archived).

Collection and basic log properties: During a live scan, the tool attempts to retrieve the logs' acceptable/trusted roots using the *get-roots* method of each available log. However, since only Google's and Let's Encrypt's logs include Cross-origin Resource Sharing (CORS) related headers in their responses (see "CORS header" in Table I), we had to disable the CORS enforcement in our browser (used to protect modern browsers) to retrieve data from the other logs. Furthermore, due to some logs having invalid server certificates we disabled TLS certificate verification. These aspects also complicate creation of secure client-side browser tools, extensions and web-pages for the utilization of CT.

To compare root stores, the tool removes duplicates from JSON root lists returned by *get-roots*, leaving only distinct certificates (see "Distinct certificates" and "Duplicated certificates" in Table I). To identify equivalent root stores, we concatenate unique root hashes in ascending order and compare the obtained fingerprints (see "Distinct list"). Over time we have seen an increase in the number of logs (54→57), but a decrease in the number of distinct lists (18→15).

To determine status and expiry constraints, for each log, we performed several test submissions of trusted (expired/non-expired) certificate chains. We then determined whether the

submissions were successful ("+") or unsuccessful ("−") due to an expiration rejection criteria or some other reason. As expected, logs with a year in their name reject certificates that do not expire within a specified period. For the other logs, our test submissions were successful with the exception for two logs (Aviator and Venafi Gen2 CT) in Dec. 2018 and eight logs in Oct. 2019 (Nimbus20{17-18}, Argon2017, Nessie/Yeti2018, Aviator, Venafi Gen 2 CT, WoTrus; five logs by CNNIC, GDCA, and SHECA are now offline or have an expired certificate). Some logs reject expired certificates (see "Reject expired" in Table I); logs with expiration constraints may be listed as "usable" (e.g. Argon2018), but by rejecting expired certificates they essentially become frozen. In some cases, DigiCert Log Server 2 rejects submissions due to rate limitations.

IV. High-level Comparisons

We next characterize high-level relationships between root stores. For detailed, interactive exploration of these relationships, we encourage the use of our tool.

Vendor coverage: Figure 2 (top part) shows the fraction of roots trusted by Mozilla, Microsoft and Apple that are not covered by the logs (using logarithmic scale). A closer look reveals interesting observations. First, the root stores are significantly different both between and within individual operators. As expected, for individual operators, the vendor coverage is typically (but not always) highest for trusted logs and smallest for testlogs and other non-trusted logs (e.g., Cirrus). Second, motivated by the CT standard suggesting that a list of acceptable root certificates "might usefully be the union of root certificates trusted by major browser vendors" [21], many logs have increased their coverage to 100% for at least one or two vendors. (Changes are shown in Table I and support for our "motivated by" claims are based on our operator survey in Section V-A.) Third, despite this increase, every log except Dodo (72 added roots), is missing some fraction of roots trusted by Mozilla, Apple or Microsoft. Fourth, on average, the root stores of Mozilla and Apple are better covered than Microsoft's. Fifth, DigiCert, has the lowest vendor coverage of the five trusted log operators, and do not appear to have adjusted its root stores to match the changes made by the vendors.

Trusted vs. non-trusted logs: Although Apple's list of trusted logs differs from that of Google (see respective "Log Program" in Table I), the differences are small and mostly related to logs with expiration dates in 2017-2018. We also illustrate this in Figures 2 and 4, where we differentiate between "trusted" logs (black text), "non-trusted" logs (red text) and the production logs (e.g., Nimbus 20{17-23}) that are trusted for years 20{19-23} but no longer for prior year (e.g., 20{17-18}). Most importantly, both Apple and Google rely exclusively on logs by five operators: Cloudflare, DigiCert, Google, Sectigo, and Let's Encrypt (where the last operator's logs became Google "qualified" on Oct. 7, 2019). Logs from other operators are non-trusted (i.e., rejected, retired, not listed, or read only). Furthermore, no Apple-trusted log by Google or DigiCert satisfies Apple's requirement to "trust all root CA certificates included in Apple's trust store" [6], and both Cloudflare and Let's Encrypt only do so because of recent updates to their root lists. These observations suggest that some requirements of the evolving policies are not strictly followed. Among non-trusted logs, we note many test logs (e.g., Testtube, Solera). Their root lists are smaller, mainly contain test certificates, and miss almost all roots included by major vendors.

Root store size evolution: Figure 3 shows how the root store sizes of the five major log operators have changed over time; for the most part making the root stores larger and more similar in size. We also include the timeline of a testlog (Testube, which shares root store with Cruscible and Solera 20{18-22}). Cloudflare (May 2019) and Let's Encrypt (June 2019) both made major increases to their root stores, making them more similar in size to those of Google and DigiCert. At this time, they also increased their coverage of the root stores of the major vendors. (From the survey, we learned that Cloudflare update in May 2019 was triggered by email from a CA.) Interestingly, despite Sectigo Mammoth/Sabre having the highest vendor coverage (Figure 2), they have the smallest root store. Finally, we note that during our measurement campaign the production logs had much fewer root store change events (three on average) than the testlog (41 events).

Root store (dis)similarity: Figure 4 shows the pairwise root store overlap between logs, where the overlap is calculated as $|X \cap Y| \,/\, |X|$ for each pair of logs X and Y. We note that many pairs overlap substantially and differ only by a few roots (light yellow). The largest differences are between the test logs and production logs (e.g., darker regions associated with Google Solera, Crucible and Testtube). Previously, two GDCA logs (no longer accessible) and ten DigiCert logs were "identical" after removing duplicates. Finally, Figure 5 highlights some differences observed between the vendor stores and the major Chrome/Apple-trusted logs, and how the overlaps between some of these root stores have changed. In almost all cases, the individual root stores contain unique roots. Among the vendors, Microsoft has the largest and most dis-similar root store. Among the logs, the root store of Cloudflare Nimbus became much more similar to the root stores of Google Argon and DigiCert Yeti/Nessie, who already shared most of their roots. Let's Encrypt Oak and Sectigo Mammoth/Sabre have fewer roots and smaller overlap with Argon. As a reference point, we remind the reader that Mammoth/Sabre has close to full coverage of the three vendor stores.

Root frequencies: Trustworthiness of CAs is relative. It is therefore not surprising that some roots are more frequently included than others. Figure 6 shows the frequency distribution of the observed roots seen at the time of our two main snapshots. Here, we first grouped distinct certificates by the number of Chrome-trusted logs that use them as roots, and then plotted the number of certificates in each group. The most frequent root certificate (previously included in the root lists of 27 Chrome-trusted logs and currently by 37 such logs) is the *Merge Delay Monitor Root*. Logs are required to include this

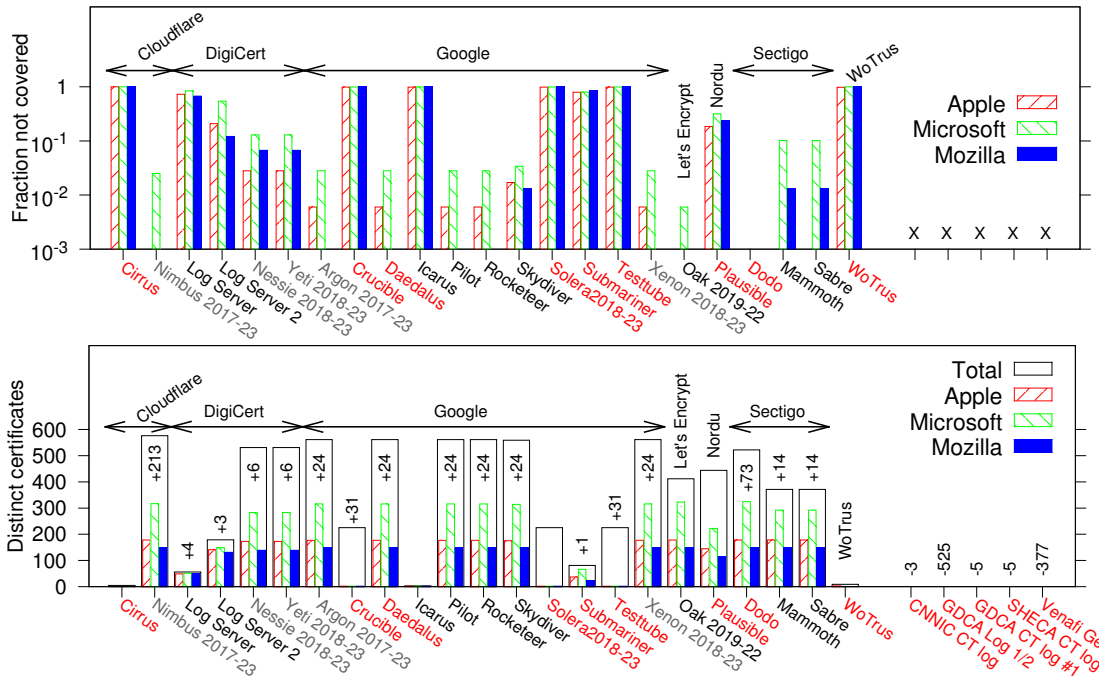


Fig. 2. Top: Fractions of vendor root stores that are not covered by the logs. Bottom: Log root stores and intersections with vendor root stores. Positive/negative numbers designate change between the initial and final measurement. Log name color: "Trusted according to Google's list of trusted logs" (black), "Not Trusted/Not included in Google's list of trusted logs" (red), "Trustworthiness depends on a year" (grey).

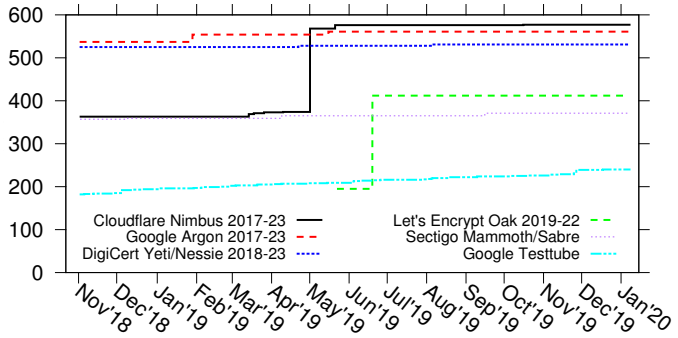


Fig. 3. Root store sizes as a function of time.

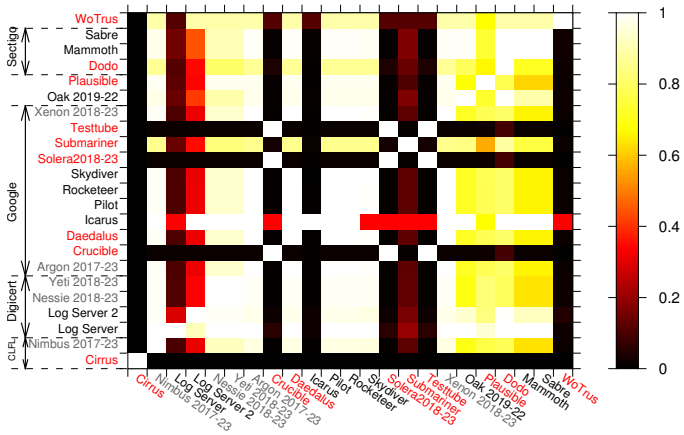
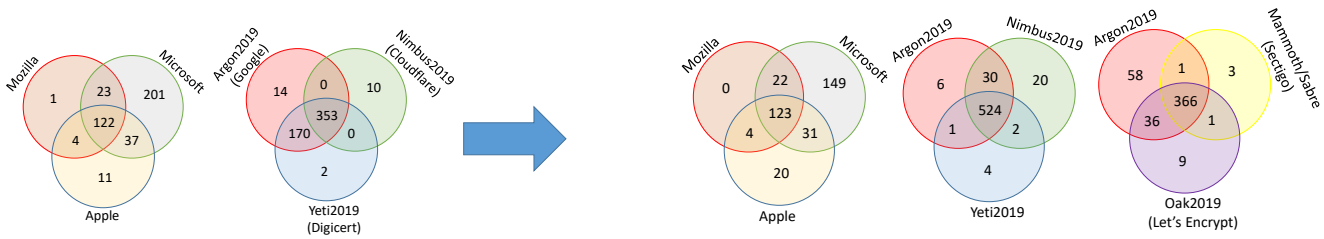


Fig. 4. Pairwise root store overlap between CT logs. $|LogX \cap LogY| / |LogX|$. Text color label: "Trusted" (black), "non-trusted" (red), depends on expiry date (grey).

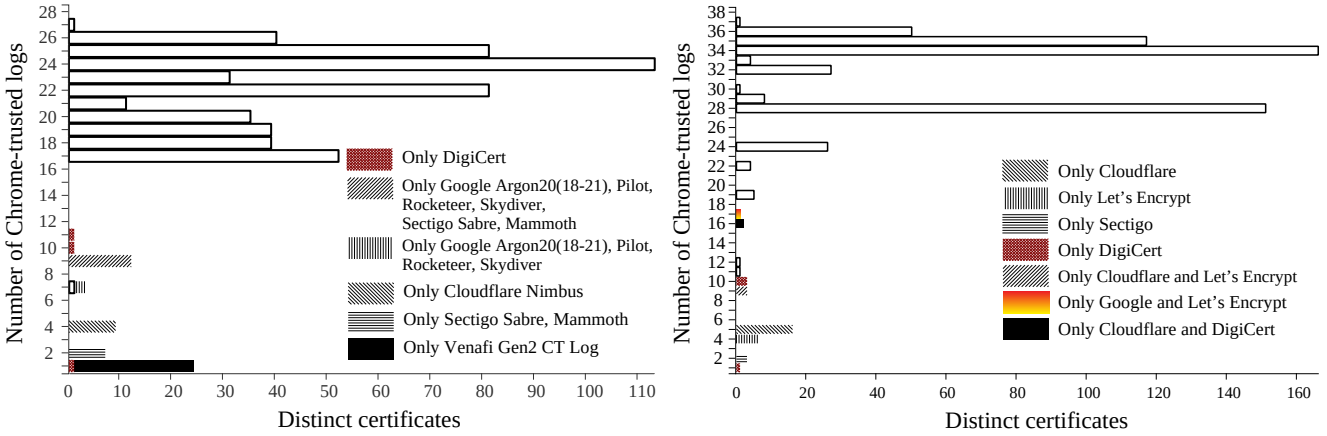
root to become Chrome-trusted, but the root is also present in most non-trusted logs. Apple's corresponding root is not as frequent. Otherwise, the certificates fall into two categories. Most of the vendor roots are covered by the majority of the Chrome-trusted logs. With the exception of two roots (covered by 11 and 12 logs in the last snapshot), the rest of the roots covered by less than 17 logs (some of which are not present in vendor stores) are included in the root lists by the logs of just one or two operators. This is not sufficiently redundant. Venafi Gen2 CT, responsible for the majority of the least frequent certificates in the Dec. 2018 dataset, is now frozen. However, most of these certificates are not self-signed roots, but intermediaries that can be chained back to already included roots.

High skew in root usage: There is a high skew in the usage of different roots and many of the roots included by the major logs are not used. Figure 7(a) shows an upper bound on the number of roots that was used by the Nimbus20{19,20} and Argon20{19,20} logs each day during the measurement period and Figures 7(b) shows an upper bound on the number of days that each root may have been used. These bounds were obtained using Censys [8] by extracting information about all certificates logged to these logs and checking which set of roots that these certificates chain back too (can be more than one). Comparing these numbers with the root store sizes of these logs (which changes from 363 to 576 and from 537 to 561, respectively, during the measurement period), we note that more than half of the roots are not used at all, and less than 113 (Argon2020) of the roots are used more than 10% of the days.



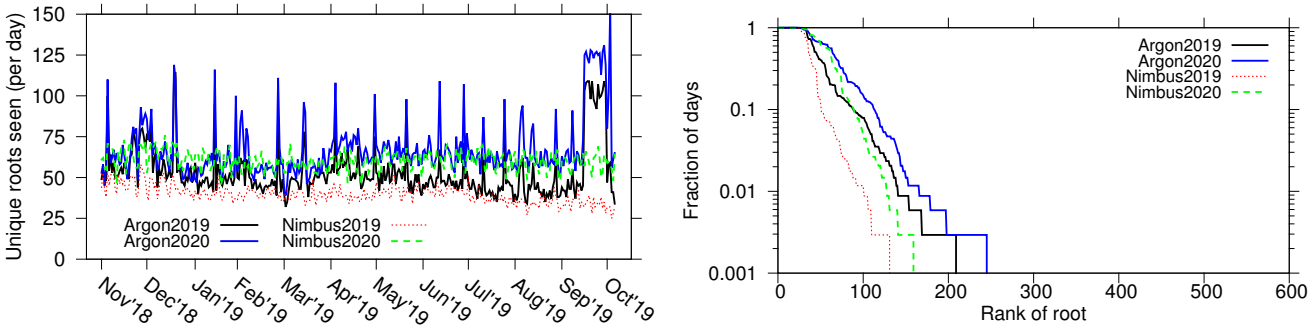
(a) Dec. 27, 2018 (b) Oct. 8, 2019

Fig. 5. Intersections of vendor root stores and example production logs.



(a) Dec. 27, 2018 (b) Oct. 8, 2019

Fig. 6. Certificates grouped by how many Chrome-trusted logs include them.



(a) Roots used each day (b) Days root used

Fig. 7. Upper bounds of individual root usage.

V. Log Management

A. Operator survey

We emailed questions (Sept. 23, 2019) about log management to the five Google/Apple trusted log operators, and got answers from all five: Google, Let's Encrypt, Cloudflare, Sectigo, and DigiCert. The questions are listed in appendix.

With exception of DigiCert, the operators create their root lists automatically, in some cases involving a manual review step before lists are promoted to the production logs. Both Google and Sectigo explicitly state that they use the union of the trusted roots from the Mozilla, Microsoft, and Apple trusted root programs. The main difference is that Google typically do not remove roots from the set of accepted roots (although they state that there is on-going discussion about potentially starting to do this), whereas Sectigo base their lists on their current crt.sh data [34]. Cloudflare have their own roots program, in which they explicitly lists root stores they cover, but they also update based on requests from CAs. Let's Encrypt also mentioned that they obtained a root from Apple (per email) to perform similar tests as Google's MMD root and that they plan to create similar tools as those used by Cloudflare (e.g., [7]).

DigiCert maintains a master file of acceptable roots that gets pushed out with every deploy. The original list for their high performance logs was created from the browser trust stores, to which they add roots included in at least one browser root store as email requests to include these come in. At some point, they plan to find and add all the roots that are acceptable by the new Google logs.

Cloudflare and Sectigo stated that (as far as they know) they have not intentionally enabled or disabled CORS, with Cloudflare noting that they currently do not set them since

they have not yet had reason to update their code (which is based on the Trillian code base) to the latest version of Trillian (which allows CORS headers), and note that such updates would involve significant changes and testing of closed-source code for their internal infrastructure.

All operators seem to agree that the process of root management is secure enough. The main comments that came up were related to more transparent root management of the logs. For example, Let's Encrypt mentioned the value of public monitors such as the one developed (and made available) in this paper and how they believe that the CT landscape could benefit from root program operators providing an easily ingestible list of root CAs trusted on various platforms. DigiCert suggested that it would be nice if log operators had a statement explaining their criteria for including new roots. Google brought up the idea (for future versions of CT) to potentially "require logs to commit changes to their set of trusted roots as a new leaf type which they log to themselves." This would make it harder to play games with root sets, and would provide a new mechanism for others to become aware of updated root sets.

B. Incidents and potential mismanagement

We have observed a number of events and behaviors that warrant comments.

Duplicated roots: Several logs respond to the *get-roots* method with root lists that include duplicates ("Duplicates" in Table I and Table II in appendix). These duplicates may have been introduced by accident due to manual log (mis)configuration, or a bug in a log server implementation. However, we have found that the misbehavior of GDCA and Sectigo logs (described next) coincided with anomalous presentation of their root stores, when duplicate roots were introduced. Moreover, the frozen Venafi Gen2 CT log also contains duplicates. We conclude that the presence of duplicated roots could be used as an indicator of log's potential (root) mismanagement. At this moment, DigiCert Log Server 2, Yeti and Nessie logs still contain duplicates. DigiCert acknowledge this as an oversight likely caused by a CA having asked for inclusion of a root that already was in the list of acceptable roots, but do not consider this a serious issue. We believe that a mechanism for transparent management of root lists would be beneficial. Such a mechanism would improve monitoring and could prevent trust-related log misbehavior. Versioning of root stores in public repositories similar to [31] could be a possible solution.

GDCA logs: On Aug. 16, 2018, GDCA Log 1 failed to incorporate two SCTs within the MMD. The SCTs were issued during an update of the root lists for Log 1 and Log 2 [23]. According to GDCA, the accident was caused by a reboot after the update. Google disqualified Log 1 and GDCA withdrew their request to include Log 2 in Chrome. Interestingly, we found that duplicates were introduced at the time of the incident and that they remained there until the logs were shut down (on Jan. 15 and Feb. 14, 2019).

Sectigo Mammoth, Sabre and Dodo: We have observed outages for Sabre and Mammoth, with up-time of Sabre going below 99%. We have also noticed that Mammoth was sporadically returning two different root lists, as *Fina Root CA* and *Hongkong Post Root CA 3* have been randomly appearing in Mammoth's root list between Dec. 6, 2018 and Jan. 2, 2019. While Sectigo answered our survey, they have yet to comment on these events.

C. Notable roots and use cases

Test roots in trusted logs: In general, you would not expect test roots in Chrome or Apple-trusted logs. However, we have found that DigiCert Log Server 2, Yeti and Nessie logs include two *DigiCert CT Test Root*s in their root lists that are not directly logged in any of the trusted logs. (While most attempts were rejected due to rate limits setup for the older DigiCert logs, to protect the log from risk of failure or missing the MMD for new entries, we could not resist the temptation, and have now logged one of these certificates [1].) DigiCert have since explained that they included some test roots for monitoring purpose. Let's Encrypt logs also include a test root – *ct-woodpecker*.

DigiNotar Root CA in Cloudflare logs: DigiNotar is a former CA that went bankrupt due to a well-known attack [29] on its infrastructure. This attack and the fact that misissued certificates could remain undetected was one of the reasons for the creation of CT [2]. Surprisingly, Cloudflare Nimbus logs include *DigiNotar Root CA* certificate in their lists of acceptable roots, years after the attack. Assuming that there still exists a party with access to its private key, this party could issue an arbitrary number of DigiNotar-signed certificates acceptable by Nimbus logs[1]. When asked, Cloudflare did not know why this root was in their root store, but have no plans to remove it either. (A closer look at the Cloudflare root policy [7] suggests that the inclusion may be due to them using the union of several OS trust stores, including some that include DigiNotar (e.g., Android Gingerbread and Honeycomb). While we have seen roots being removed, some operators appear to rather be inclusive than hinder some CAs from submitting.

The above example presents an interesting dilemma. Nowadays, Cloudflare Nimbus is the only log that allows DigiNotar certificates. It potentially could help understand or even identify the attacker if a rogue certificate is logged. However, in the case when other operators add this certificate to their lists of acceptable roots (e.g., to increase the chance that such certificates are logged), an attacker would be able to obtain enough SCTs to satisfy requirements of CT policies, and hence *may* more easily trick a client on which the DigiNotar Root CA certificate is somehow valid. To avoid such future conflicts, we suggest that non-trusted roots perhaps should be stored in separate logs; e.g., similar to how Google Daedalus (discussed

[1]While there exist many other ways to obtain numerous valid certificates for submissions (e.g., downloading certificates from other logs or tweaking existing ones), the capability to generate arbitrary many acceptable certificates could also be used for DoS attacks. We note that a successful attack on a log may interrupt issuance process of some CAs relying on that log.

below) is used to store expired certificates. Moreover, a common and homogeneous approach to the formation of root stores in logs would eliminate such outlying cases.

Cirrus: A closer look at the root stores (and the logs' content) also highlight other interesting example use cases of CT. For example, rather than storing certificates associated with the WebPKI, Cloudflare Cirrus is used with Resource Public Key Infrastructure (RPKI) [10] and its root store only contains the certificates of the five Regional Internet Registries (RIRs): AFRINIC, APNIC, ARIN, LACNIC, RIPE.

Daedalus: While the root store of Google Daedalus is equivalent to the stores of the Argon, Xenon, Pilot and Rocketeer logs, it is unique in that it only accepts expired certificates. It is not trusted by Chrome nor Apple. We believe that Daedalus provides an example of how non-trusted logs can be used to log non-trusted certificates of potential interest, which we believe could be extended to cover some of the roots that the browsers no longer trust (e.g., DigiNotar example above).

SHECA CT log 2: The skew in usage of the logs is high [18], with some logs seeing very limited use. As an extreme example, SHECA CT log 2 was idling with just two certificates in its tree between Apr. 2017 and Jan. 2019, when we successfully performed a test submission. The log's server certificate could be chained back to Mozilla's root store, but not to Apple's, meaning that by default, Apple clients were unable to establish a secure connection to the log. The log was taken offline in Mar. 2019.

VI. Related work

While a number of studies have characterized the CT landscape [18], [30], very limited results have been reported regarding their root selections. For example, Gustafsson et al. [18] provide basic root count for eleven public logs available in Dec. 2015 and list basic log properties, but primarily provide statistics (based on active and passive measurement data) of the relative CT usage among domains and CAs. Most other papers focus on logged certificates [11], [20] and on specific aspects, such as server-side use of SCTs [27], [2] and client-side performance of SCT delivery methods [27]. Amann et al. have studied and compared the adoption of several technologies (including CT) that strengthen the WebPKI [2]. Privacy issues of CT are considered by Eskandarian et al. [9]. A longitudinal study on CAs, log operators, and CT deployment on servers is presented by Scheitle et al. [30]. In the study, they also apply CT for the detection of phishing domains and highlight the use of CT by third-parties for malicious purposes. Stark et al. [33] study the adoption of CT and estimate the compliance to Google's CT policy using client-side Chrome telemetry. In contrast to prior work, our focus is on the root stores of the logs and their relationship to the root stores of major vendors and CT policies.

VII. Conclusion

In this paper, we presented a first characterization of the emerging root store landscape and the *CT Root Explorer* tool for interactive analysis thereof. We overviewed CT logs and analyzed their root stores relative to the stores of Apple, Microsoft and Mozilla. We have monitored and summarized the properties of 57 CT logs, including all available logs from Google's and Apple's lists of known logs, and observed how the root stores change over time.

The landscape of CT is evolving: the technology is highly established and is commonly used by the *majority* of CAs and clients on the Internet, which makes it strategically important; software vendors introduce and change their CT policies, new logs are established regularly, while the recent introduction of Cloudflare Cirrus extended the use of CT to ResourcePKI. Over our measurement campaign root stores have became larger and increased their coverage of the roots used by the major vendors.

To directly measure the status of the available CT logs, we have performed a number of test submissions, and by calculating frequencies of root certificates in logs, we have found a number of roots that are included in the root lists by just one or two log operators. Surprisingly, Cloudflare Nimbus logs include a compromised DigiNotar root certificate in their root lists. Multiple logs are considered Apple/Google-trusted despite violating corresponding policies. We have discovered that all CT logs (except Google's and Let's Encrypt's) do not specify cross-origin headers in their HTTP responses, which obstructs access to the logs using JavaScript in modern browsers.

Overall, we have observed that some WebPKI roots trusted by major software vendors are not sufficiently covered by the CT logs; Apple and Google rely on the CT backbone that is comprised of just five actively logging operators: Cloudflare, DigiCert, Google and Sectigo (all of these operators are US-based). Several logs have already been disqualified, while Sectigo's trusted production logs (Mammoth/Sabre) have been seen to struggle with their up-time requirements. Moreover, we have found logs with duplicates in their root lists and that an accident with GDCA Log 1 and outages of Sectigo Mammoth coincide with anomalous presentation of their root stores. One other misbehaving log which is currently in read-only state (Venafi CT Gen2) has also been presenting root stores with duplicated entries in it. Finally, we argue that management of CT policies and logs' root stores must be performed in a more careful, timely and transparent manner.

Acknowledgment

We are very thankful to the log operators who responded to our survey and provided additional insights to our log specific observations. This work was supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

References

[1] DigiCert submission (2019), https://crt.sh/?sha256=6940c8f47652fe06432c231886808089c317ecd0503aa60383395e4f22fe8656 Last accessed: Jan. 2020

[2] Amann, J., Gasser, O., Scheitle, Q., Brent, L., Carle, G., Holz, R.: Mission Accomplished?: HTTPS Security After DigiNotar. In: Proc. IMC (2017)

[3] Apple: List of available trusted root certificates in iOS 12, macOS 10.14, watchOS 5, and tvOS 12 (2019), https://support.apple.com/en-us/HT209144, last accessed: Jan. 2020
[4] Apple: List of available trusted root certificates in iOS 12.1.3, macOS 10.14.3, watchOS 5.1.3, and tvOS 12.1.2 (2019), https://support.apple.com/en-us/HT209501, last accessed: Jan. 2020
[5] Apple: Apple's Certificate Transparency log program (2020), https://support.apple.com/en-us/HT209255, last accessed: Jan. 2020
[6] Apple: Apple's Certificate Transparency policy (2020), https://support.apple.com/en-us/HT205280, last accessed: Jan. 2020
[7] Cloudflare: CFSSL's CA trust store repository (2020), https://github.com/cloudflare/cfssl_trust, last accessed: Jan. 2020
[8] Durumeric, Z., Adrian, D., Mirian, A., Bailey, M., Halderman, J.A.: A search engine backed by Internet-wide scanning. In: Proc. ACM CCS (2015)
[9] Eskandarian, S., Messeri, E., Bonneau, J., Boneh, D.: Certificate Transparency with Privacy. In: Proc. PETS (2017)
[10] Fleury, J., Poinsignon, L.: RPKI and BGP: our path to securing Internet routing (2018), https://blog.cloudflare.com/rpki-details/, last accessed: Jan. 2020
[11] Gasser, O., Hof, B., Helm, M., Korczynski, M., Holz, R., Carle, G.: In Log We Trust: Revealing poor security practices with Certificate Transparency logs and Internet measurements. In: Proc. PAM (2018)
[12] Google: Root Certificate Policy, https://www.chromium.org/Home/chromium-security/root-ca-policy, last accessed: Jan. 2020
[13] Google: Certificate Transparency Open Issues (2018), https://bugs.chromium.org/p/chromium/issues/list?q=component:Internals\%3ENetwork\%3ECertTrans, last accessed: Jan. 2020
[14] Google: Chromium Certificate Transparency Policy (2019), https://github.com/chromium/ct-policy, last accessed: Jan. 2020
[15] Google: The list of CT logs that are compliant with Chrome's CT policy (or have been and were disqualified), and are included in Chrome (2019), https://www.gstatic.com/ct/log_list/v2/log_list.json, last accessed: Jan. 2020
[16] Google: HTTPS encryption on the Web. Transparency Report (2020), https://transparencyreport.google.com/https/certificates?hl=en, last accessed: Jan. 2020
[17] Google: The list of all known and announced CT logs (2020), https://www.gstatic.com/ct/log_list/v2/all_logs_list.json, last accessed: Jan. 2020
[18] Gustafsson, J., Overier, G., Arlitt, M., Carlsson, N.: A first look at the CT landscape: Certificate Transparency logs in practice. In: Proc. PAM (2017)
[19] Korzhitskii, N.: Certificate Transparency Root Explorer (2019), https://nikita-kun.github.io/certificate-transparency-root-explorer/
[20] Kumar, D., Wang, Z., Hyder, M., Dickinson, J., Beck, G., Adrian, D., Mason, J., Durumeric, Z., Halderman, J.A., Bailey, M.: Tracking certificate misissuance in the wild. In: Proc. IEEE S&P (2018)
[21] Laurie, B., Langley, A., Kasper, E.: Certificate Transparency. RFC 6962 (June 2013)
[22] Leavitt, N.: Internet Security under Attack: The Undermining of Digital Certificates. Computer **44**(12), 17–20 (2011)
[23] McMillion, B.: Un-incorporated SCTs from GDCA1. Certificate Transparency Policy, Google Groups (2018), https://groups.google.com/a/chromium.org/forum/#!topic/ct-policy/Emh3ZaU0jqI, last accessed: Jan. 2020
[24] Microsoft: Microsoft trusted root certificate program: Participants (2019), https://aka.ms/trustcertpartners, last accessed: Oct. 2019
[25] Mozilla: PKI:CT (2014), https://wiki.mozilla.org/PKI:CT, mozilla Wiki, Last accessed: Jan. 2020
[26] Mozilla: Mozilla included CA certificate list (2020), https://wiki.mozilla.org/CA/Included_Certificates, last accessed: Jan. 2020
[27] Nykvist, C., Sjöström, L., Gustafsson, J., Carlsson, N.: Server-side adoption of Certificate Transparency. In: Proc. PAM (2018)
[28] O'Brien, D.: Certificate Transparency Enforcement in Chrome and CT Day in London (2018), https://groups.google.com/a/chromium.org/d/msg/ct-policy/Qqr59r6yn1A/2t0bWblZBgAJ, last accessed: Jan. 2020
[29] Prins, J.R., Cybercrime Business Unit: DigiNotar Certificate Authority Breach 'Operation Black Tulip'. Fox-IT (Nov 2011)
[30] Scheitle, Q., Gasser, O., Nolte, T., Amann, J., Brent, L., Carle, G., Holz, R., Schmidt, T.C., Wählisch, M.: The rise of Certificate Transparency and its implications on the Internet ecosystem. In: Proc. IMC (2018)
[31] Sectigo: CTLogs-AcceptedRoots. GitHub repository (2019), https://github.com/sectigo/CTLogs-AcceptedRoots, last accessed: Jan. 2020
[32] Shinder, T.W.: Certificate Transparency. Microsoft Azure Security and Compliance (2018), https://blogs.msdn.microsoft.com/azuresecurity/2018/04/25/certificate-transparency/, last accessed: Jan. 2020
[33] Stark, E., Sleevi, R., Muminovic, R., O'Brien, D., Messeri, E., Felt, A.P., McMillion, B., Tabriz, P.: Does Certificate Transparency Break the Web? Measuring Adoption and Error Rate. In: Proc. IEEE S&P (2019)
[34] Stradling, R.: Root certificates accepted by Sectigo's CT logs (2020), https://github.com/sectigo/CTLogs-AcceptedRoots, last accessed: Jan. 2020

APPENDIX

Our survey contained seven questions (plus some log-specific questions based on our findings about the operators logs). These questions are listed next:

- Do you setup the lists of acceptable roots for your logs manually or automatically?
- How is the set of acceptable roots formed/updated? (I.e., what is your policy for managing the list of acceptable root certificates?)
- How often do you update your list(s) of acceptable roots?
- Are there any procedures for root store verification and provision in place?
- (If applicable) Some logs do not explicitly ship cross-origin (CORS) headers with their responses, which prevents browsers from querying the logs using javascript. Is this intentional? And if so, what is the reason?
- In general, do you consider the process of root management transparent and secure enough?
- (If applicable) What kind of improvements would you suggest to increase the security and transparency of the logs and to avoid future misconfigurations?

Tables II and III list the specific roots that were duplicated in the JSON root lists of some of the logs, and the logs that contained these duplicates, as observed on Dec. 27, 2018, and Oct. 8, 2019, respectively.

TABLE II
LOGS WITH DUPLICATES IN ROOT LISTS (DECEMBER 27, 2018)

Logs	Duplicated Root
DigiCert Log Server 2 DigiCert Yeti20* DigiCert Nessie20*	Atos TrustedRoot 2011
GDCA Log 1	DigiCert Trusted Root G4
GDCA Log 2	Merge Delay Monitor Root
Venafi Gen2 CT log	Buypass Class 2 Root CA Buypass Class 3 Root CA TWCA Root Certification Authority Certum Trusted Network CA T-TeleSec GlobalRoot Class 3 Entrust Certification Authority - L1E

TABLE III
LOGS WITH DUPLICATES IN ROOT LISTS (OCTOBER 8, 2019)

Logs	Duplicated Root
DigiCert Log Server	Atos TrustedRoot 2011 (1)
DigiCert Log Server 2	Atos TrustedRoot 2011 (2)
DigiCert Yeti20*	Atos TrustedRoot 2011 (2)
DigiCert Nessie20*	VRK Gov. Root CA