2019 IEEE Symposium on Security and Privacy

Does Certificate Transparency Break the Web? Measuring Adoption and Error Rate

Emily Stark[1], Ryan Sleevi[1], Rijad Muminović[2], Devon O'Brien[1], Eran Messeri[1],
Adrienne Porter Felt[1], Brendan McMillion[3], Parisa Tabriz[1]
[1]*Google,* [2]*University of Sarajevo,* [3]*Cloudflare*
[1]estark,rsleevi,asymmetric,eranm,felt,parisa@chromium.org, [2]rmuminovic1@etf.unsa.ba, [3]brendan@cloudflare.com

***Abstract*—Certificate Transparency (CT) is an emerging system for enabling the rapid discovery of malicious or misissued certificates. Initially standardized in 2013, CT is now finally beginning to see widespread support. Although CT provides desirable security benefits, web browsers cannot begin requiring all websites to support CT at once, due to the risk of breaking large numbers of websites. We discuss challenges for deployment, analyze the adoption of CT on the web, and measure the error rates experienced by users of the Google Chrome web browser. We find that CT has so far been widely adopted with minimal breakage and warnings.**

Security researchers often struggle with the tradeoff between security and user frustration: rolling out new security requirements often causes breakage. We view CT as a case study for deploying ecosystem-wide change while trying to minimize end user impact. We discuss the design properties of CT that made its success possible, as well as draw lessons from its risks and pitfalls that could be avoided in future large-scale security deployments.

***Index Terms*—Web PKI, HTTPS, Certificate Transparency, usable security**

I. Introduction

In 2011, a widely trusted certificate authority (CA) improperly issued an HTTPS certificate for Google domains. This misissued certificate was used to target Iranian internet users in a man-in-the-middle attack [1]. The certificate was revoked and the offending CA was removed from client trust stores, but the incident demonstrated the danger of improper certificate issuance and the need to strengthen the web PKI against attacks.

Certificate Transparency (CT) is an emerging system that facilitates the discovery of certificates that might be used in attacks. CT improves the web PKI by allowing domain owners to discover unexpected certificates issued for their domains and by allowing the public at large to discover suspicious or improper CA issuance practices. With CT, certificates are recorded in publicly-auditable, append-only logs. Clients can choose to trust certificates only when they are accompanied by proofs that they have been publicly logged.

The full deployment of CT is a dramatic change to the HTTPS ecosystem. Eventually, to achieve the full security properties of CT, web browsers will enforce CT for all publicly trusted certificates: that is, web browsers will not accept any public certificate unless it has been logged in CT logs. Some browsers enforce CT partially, but none fully enforces CT for all certificates yet. In practice, browsers would not roll out full CT enforcement all at once, for fear of causing widespread breakage. Users would see certificate warnings on any website that did not properly implement CT. Frequent warnings can cause warning fatigue and get in the way of users doing important tasks on the web [2], [3].

In this paper, we explore whether CT's deployment has been successful so far. Have browsers, CAs, and websites deployed it with low error rates for users? What has contributed to CT's successes and failures, and what can the security community learn from it? To study these questions, we measure CT adoption and health from several perspectives and over time. We find that CT has been widely deployed, with over 60% of the web's HTTPS traffic now supporting CT, and that CT rarely causes warnings or breakage, even when new CT requirements go into effect. When breakage does occur, it is often due to bugs or misconfigurations in how certificate authorities (CAs) implement CT. When users encounter these CT errors, they engage in unsafe behaviors, such as bypassing the warning or switching browsers rather than heed the warning's security advice. This underscores the need to keep the error rate low as CT rolls out more broadly.

We attribute the low error rate and wide deployment of CT to three main characteristics of the system. First, CT adoption does not require individual site owners to take action; a relatively small number of CAs can do the legwork to deploy CT across much of the web. Second, CT enforcement can be rolled out in stages, gradually ramping up adoption and flushing out problems Finally, web browsers are free to determine the specifics of how they enforce CT. This flexibility mitigates many of the security and operational risks that exist in the budding CT ecosystem.

We also identify hurdles in CT deployment that might present challenges for future systems. The rollout of CT has been largely driven by a small number of major ecosystem players, and deployment might not have been successful if only smaller players had been invested in its success. Moreover, CT support has lagged in browsers other than Google Chrome: CT represents a substantial investment for browser vendors, as well as log operators and CAs. Gaining broader adoption among browsers is crucial for CT's success.

A. Contributions

Our primary contributions are:

DOI 10.1109/SP.2019.00027

- We measure how widely CT has been adopted across the web and explore factors that have contributed to its successful adoption.
- We explore how often users experience CT-related breakage and how they react to it. We find that breakage rates are low, and we identify aspects of CT that minimized the negative impact. However, we also find that users tend to behave unsafely when faced with current CT warnings.
- Using anecdotes from large CT deployments, we identify risks in CT that could cause disruption to end users as CT enforcement rolls out more broadly.
- We discuss hurdles that might pose challenges for similar systems in the future.

II. Background

This section gives an overview of HTTPS and the web PKI, as well as CT and its current state of deployment.

A. HTTPS and the web PKI

HTTPS encrypts and authenticates web traffic to protect its confidentiality and integrity against network attackers. When HTTPS is in use, the client performs a handshake with the server after setting up a TCP connection. The handshake establishes an encrypted and authenticated channel over which the client and server can exchange HTTP requests and responses.

During the handshake, the client validates a certificate, which authenticates the server in the web PKI. A server owner can obtain a certificate from any of a number of certificate authorities (CAs), which use several validation methods to verify the server's identity.

Clients maintain a *trust store*, a list of CAs that they trust. To validate a server certificate, the client attempts to build a chain from the provided certificate to a CA in its trust store. The client also performs other checks, such as making sure that the certificate is not expired.

If the client cannot validate the certificate, then the connection will fail because the client cannot verify the server's identity. Web browsers show full-page error warnings when they encounter a certificate error for the main resource of a page load. When a browser cannot validate the certificate of a subresource, such as a script or image, then it will often simply fail to load the resource. When a subresource fails to load, the page might appear to be partially or fully broken to the user. For example, if a critical JavaScript library doesn't load, the page's main functionality may not work.

CAs are usually trusted to issue certificates for any website. If a single CA is compromised or malicious, the attacker can mount man-in-the-middle attacks on a large number of high-value websites if they can intercept traffic to those websites.

We refer the reader to [4] and [5] for more detailed background about the web PKI.

B. Certificate Transparency

CT aims to protect users from mistakenly or maliciously issued certificates by ensuring that all certificates are logged in publicly-auditable, append-only logs [6]. Domain owners can monitor CT logs to discover improperly issued certificates for their domains. Moreover, CT logs are open for public auditing and monitoring: anyone can monitor for suspicious certificates or CA misbehavior. When a suspicious certificate is discovered in a CT log, the domain owner can request that the issuing CA revoke the certificate or, in the case of a misbehaving CA, the CA can be removed from client trust stores.

CT can be described in three parts: logging, SCT validation, and monitoring/auditing.

Logging. Anyone can run a CT log, which is an auditable Merkle tree of certificates. The log operator notifies browsers of the existence of the log, signaling its availability to the CT ecosystem. Similarly, anyone can submit a certificate to a CT log. Certificates are usually submitted by the CA shortly before or after issuance, but certificates are also often submitted by research scanners, web crawlers, domain owners, and others.

When a log receives a certificate, it replies with a Signed Certificate Timestamp (SCT). An SCT is a verifiable promise, signed by the log's private key, indicating that the log commits to incorporate the certificate in its public log.

SCT validation. SCTs are offered to clients with a certificate to indicate that the certificate has been or will shortly be publicly logged. SCTs can be delivered in three ways:

- Embedded in the certificate. With this method, a site owner does not need to do anything to support CT besides obtaining a certificate from a CA that embeds SCTs.
- In a TLS extension, provided as part of the HTTPS connection setup. Site owners can log their own certificates and provide SCTs using this method even if their CAs do not support CT.
- In a stapled OCSP response, which is a statement of non-revocation signed by a CA and delivered by the server to the client [7]. With this method, site owners do not have to do their own logging, but the server must support OCSP stapling and the CA must provide SCTs in their OCSP responses.

Web browsers maintain lists of CT logs, identified by public keys, that they recognize. Typically a browser will only recognize logs that meet certain availability and correctness requirements. When receiving a SCT via one of the above methods, the browser validates the signature to check that the SCT comes from one of its recognized logs. The browser might also have a *CT policy*, requiring that a certain number of SCTs or that a certain set of logs are represented, as described further in Section II-C.

The eventual goal of CT is that no public certificates are accepted as valid unless they are accompanied by SCTs to prove that they are or will soon be publicly logged.

Monitoring/auditing. CT logs expose a REST API that allows anyone to monitor the certificates that have been logged and audit that the log is behaving properly. Today, typical monitors and auditors include security researchers, browser vendors, and domain owners (who monitor specifically for suspicious certificates for their own domains). Auditors can request an *inclusion proof* that the certificate represented by a given SCT has in fact been logged. They can also request a

consistency proof to verify the append-only property of the log and check that different views presented by a log are consistent with each other. A more detailed description of monitoring and auditing APIs can be found in [8].

C. CT deployment in browsers

The CT standardization process began in 2012, but it is only recently beginning to see widespread support among web browsers and websites. Web browsers cannot enforce CT all at once for fear of breaking websites, and websites usually must be motivated to support CT (through browser enforcement requirements or other means) before they will deploy it.

Major web browsers are in various stages of implementing support for CT. No web browser yet fully enforces CT by requiring SCTs for all certificates. Mozilla Firefox has a preliminary implementation of SCT validation for gathering telemetry [9], but it is disabled by default. Apple (who makes the Safari browser) has announced plans to begin enforcing CT for certificates issued after October, 2018 [10]. Microsoft has expressed support for CT but has not announced plans to enforce CT in its Edge or Internet Explorer browsers [11].

In this paper, we focus on Google Chrome, which currently has the most advanced CT implementation of major web browsers. As of July 2018, Chrome requires CT compliance for a subset of certificates, detailed in Section II-C2.

1) Chrome CT policy: When CT is being enforced, Chrome requires SCTs which comply with a policy, known as the *Chrome CT policy* [12]. The policy requires a certain number of SCTs, ranging from two to five depending on the age of the certificate and how the SCTs are delivered. The policy also contains a *log diversity* clause, requiring the server to present at least one SCT from a log operated by Google and one operated by another entity.

By requiring multiple SCTs from diverse logs, the policy ensures that an attacker must compromise or collude with multiple entities to "hide" a malicious certificate. That is, the attacker must be able to present SCTs with valid signatures from multiple logs, yet stop all those logs from actually logging the certificate. The redundancy and diversity requirements also provide resilience in the event that Chrome removes trust in a log due to misbehavior: certificates with SCTs from a disqualified log may still continue to comply with the policy.

When CT compliance is required but a site fails to comply, Chrome shows the error UI in Fig. 1.

2) Stages of enforcement in Chrome: To encourage deployment without imposing widespread breakage, Chrome CT enforcement has rolled out in several stages (Fig. 3).

Extended Validation requirement. Since early 2015, Chrome requires compliance with the CT policy to show the Extended Validation (EV) certificate UI. The EV UI, shown in Fig. 2, is shown for a particular kind of certificate issued with extra validation checks [13]. When the certificate is not CT-compliant, Chrome removes the legal entity name and falls back to the default UI shown for regular HTTPS certificates.

In response to this requirement, most CAs began embedding SCTs in EV certificates. Chrome also compiled a whitelist of

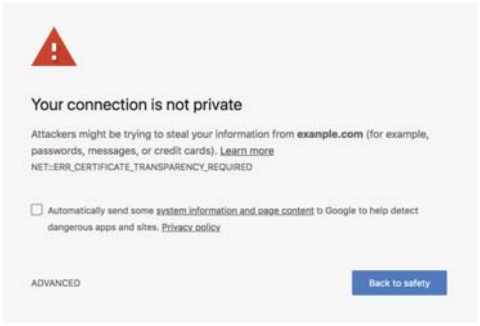

Fig. 1. The Chrome UI shown when a website is required to comply with the CT policy but fails to do so.

Fig. 2. The Chrome Extended Validation certificate UI (top) and default UI shown for regular HTTPS certificates (bottom).

EV certificates that were logged but didn't have embedded SCTs. The CT requirement was waived for certificates in this whitelist, to avoid removing the EV UI for existing EV certificates as long as they were publicly logged.

Individual CAs. Chrome also requires CT compliance for particular CAs that have misissued certificates in the past [14]. When CT enforcement is rolled out for a CA, Chrome generally requires CT compliance for only newly issued certificates, to avoid breaking existing certificates.

Site opt-in. Since September 2017, Chrome allows websites to opt in to CT enforcement via an `Expect-CT` HTTP header [15]. This feature is used by high-value websites that want the security benefits of CT even though Chrome does not require them to be CT-compliant by default. After Chrome observes the header, the browser remembers the opt-in for a configurable period of time, and requires CT compliance for connections to that website.

New certificates. Starting in late July 2018, Chrome enforces CT for all certificates issued after April 30, 2018 [16].

III. Methodology

To examine the adoption and breakage rate of CT, we analyze data from several sources.

A. Chrome usage metrics

We use browser usage metrics to assess:

- CT adoption rates, by measuring how much web traffic is served with valid SCTs. Browser metrics give more insight into CT adoption patterns than, for example, passive network measurements. The browser perspective is global and further lets us discern more structure about CT usage (for example, CT adoption on main-frame requests versus subresources, or on HTTPS connections versus requests).

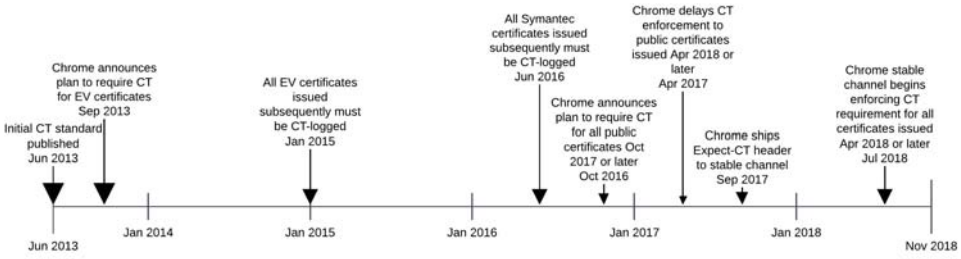

Fig. 3. Major CT enforcement milestones in Chrome.

- End user impact, primarily by measuring the rate of breakage that users see as a result of CT enforcement.

Chrome is a popular web browser with over two billion installs [17]. To assess CT's usability and adoption, we use data from Chrome's user metrics program, on the Stable channel[1]. Metrics are collected in the form of enums, booleans, counts, and times. In this paper, we use: the count of SCTs per connection and page load; an enum of CT compliance status per connection, HTTP request, and EV certificate validation; an enum of the error code each time a certificate warning is shown; and the times to verify HTTPS certificates and SCTs.

Metrics reports include client information such as operating system and country, but they do not include personal characteristics like age or gender. Chrome's user metrics program is enabled by default, but users can choose to opt out at installation time or in browser settings. (Prior to October 2016, the metrics program was disabled by default.)

B. Certificate error reports

To investigate the websites that cause the most CT-related errors in Chrome and understand how users react to them, we use a separate Chrome telemetry system that is specifically designed for gathering data about certificate errors. These error reports contain more detailed information than can be collected via the usage metrics described in Section III-A. For example, the reports contain the hostname of the website on which the error occurred, the full certificate chain, and whether the user chose to bypass the warning. Because this information can be privacy-sensitive, it is only collected on an opt-in basis. Users can opt in and out of the data collection with a checkbox on the certificate error UI shown in Fig. 1. Prior work describes the characteristics of this dataset in detail [18].

C. Server support

Browser usage statistics are heavily weighted towards a relatively small number of extremely popular websites, and do not give us fine-grained information about which sites have adopted CT. For a broader view, we also study CT adoption in terms of server support. We use three different lists of websites to measure CT adoption on the head and long-tail of the web.

[1]The Stable channel is considered the most representative for measurement purposes because it has the largest set of users and is the default release channel for installation. See https://www.chromium.org/getting-involved/dev-channel for more details.

To determine CT compliance on these websites, we analyze data produced by Googlebot (Google's web crawler). When crawling an HTTPS URL, Googlebot records the certificate and SCTs, if any. It also records the source of each SCT (TLS extension, OCSP response, or embedded in the certificate).

We built an analysis tool that consumes Googlebot records from January 2018 for each website. If the website responded with a valid certificate over HTTPS, the tool validates each SCT seen for the domain, checking that it comes from a log trusted by Chrome and that the signature is valid. It checks the certificate and SCTs against the Chrome CT policy, which is described in Section II-C. If multiple certificates and corresponding SCTs were observed for a domain, we consider the domain to be CT-compliant only if all observed certificates were CT-compliant. This approach yields a conservative estimate of CT adoption, because it does not count sites as compliant if they were only sometimes compliant.

Note that we only consider the main resource of each website, and do not analyze the CT compliance of each subresource that the website loads. This is a standard limitation in measurement research based on server scans. Analyzing the CT compliance of subresources is an interesting avenue for future work, though we do include information about subresources in data based on Chrome user metrics.

Below we describe the lists of websites that we analyze.

1) Alexa Top 10,000: The Alexa Top 10,000 contains the top 10,000 ranked domains from traffic estimates based on Alexa toolbar users and website analytics scripts [19]. We requested the Alexa list on February 5, 2018.

The Alexa list is composed of domains, and it rolls up subdomains into one domain label below the effective TLD. For example, traffic for `https://a.example.test`, `https://a.b.example.test`, and `http://example.test` is all aggregated under a single ranking for `example.test`. (Exceptions are made for some domains like `blogspot.com` which host different personal webpages as subdomains.)

2) Chrome User Experience Report: The Chrome User Experience Report is a public dataset of user experience measurements on a sample of websites [20]. The report aims to provide a broad snapshot of the web from the perspective of Google Chrome users. The list of websites is chosen from the browsing traffic of Chrome users who have opted in to share aspects of their browsing data with Google. The list is

available as a public BigQuery project. In this paper, we use the list of 1,939,945 websites provided in the December 2017 report[2]. The report includes a scheme and full hostname for each website, and does not include rankings.

3) HTTP Archive: For another view of CT adoption in the long tail, we use the domains scanned by the HTTP Archive, a project that records a view of various performance and functional characteristics of the web [21]. The sites scanned by the HTTP Archive are derived from the Alexa million[3], which is commonly used to represent the web's long tail. We fetched the HTTP Archive's list of sites scanned on January 15, 2018 from its BigQuery table[4], a total of 458,969 websites.

D. CT logs

We analyze the contents of CT logs for patterns of certificate issuance that affect end users, particularly related to Chrome's CT enforcement for EV certificates. We use a July 6, 2018 snapshot from a pipeline that ingests all the contents of 36 well-known CT logs[5].

E. Chrome product help forums

To study how users react to CT errors, we reviewed CT-related posts in the Chrome product help forum[6]. We reviewed the 75 public threads matching a search term for the error code that Chrome displays for CT errors (`ERR_CERTIFICATE_TRANSPARENCY_REQUIRED`). We coded the threads using a general inductive approach [22]. After reviewing all the threads to develop a codebook of 18 labels, two researchers independently coded each thread, applying one or more labels from the codebook to each thread. The set of labels is listed in the Appendix. To measure inter-rater reliability, we computed a Kupper-Hafner statistic [23] of 0.63, which is considered "substantial" agreement [24].

This dataset is limited to a small self-selected set of users, and it lacks demographic information. Nevertheless, the help forum threads yield several interesting and naturalistic case studies illustrating how users react to CT errors.

IV. Adoption

In this section, we measure the current state of CT adoption and examine how and why it has grown over time. We hope to see gradual growth in CT adoption because it encourages the CT ecosystem to mature without widespread disruption to end users. Gradual adoption flushes out problems early on before they cause widespread negative impact. For example, if a CA begins embedding SCTs in certificates before a browser begins requiring them to do so, the CA may find and fix problems in their implementation before those problems result in certificate errors for users. We analyze why websites have adopted CT, and we find that CT adoption is largely driven by browser requirements and the support of major ecosystem players.

A. Adoption measured from Chrome

1) Current adoption: The majority of HTTPS traffic is currently CT-compliant. We consider Chrome metrics for the week ending February 1, 2018, and we measure requests and connections. Multiple HTTP requests can share the same connection. Of HTTP requests during this time period for which CT compliance was evaluated[7], 71.1% were CT-compliant. 63.2% of HTTPS connections were CT-compliant.

2) Growth over time: CT adoption grew dramatically in 2015 and 2016, then leveled off and recently began growing again. Fig. 4 shows the historical trend in Chrome traffic that supports CT. We consider two metrics: main-frame HTTPS page loads and HTTPS connections. The former represents a lower-bound on user-visible impact when full CT enforcement rolls out. The latter represents failures that might or might not be visible to end-users when full CT enforcement rolls out. Users will probably not suffer for a noncompliant connection that loads a tracking pixel, but a connection that loads a core Javascript library could easily break an entire webpage, resulting in a poor user experience.

For both of these metrics, we do not have historical data for CT compliance, but rather for the number of SCTs. A connection could have multiple SCTs yet still not comply with Chrome's CT policy (for example, because the logs represented by the SCTs do not meet Chrome's requirement). However, current data suggests that this failure mode is quite rare, as shown by the small number of sites in Table I that serve SCTs but are not CT-compliant. We therefore consider the number of valid SCTs to be a reasonable proxy metric.

Fig. 4 illustrates how Chrome's stages of CT enforcement have driven CT adoption. The beginning of the graph shows a small but steady rise in adoption among main-frame HTTPS page loads, likely corresponding with CT enforcement for EV certificates (discussed further in Section V-B). The bump in mid-2016 could be driven by Chrome's enforcement of CT logging for new certificates issued by Symantec-owned CAs. Finally, though CT adoption remained steady for much of 2017 and the beginning of 2018, a recent increase can be seen towards the end of the graph's time period, likely due to Chrome's upcoming CT enforcement for certificates issued after April 30, 2018.

[2] `https://bigquery.cloud.google.com/table/chrome-ux-report:all.201712`

[3] `http://httparchive.org/about.php#listofurls`

[4] `https://github.com/HTTPArchive/httparchive/blob/master/docs/bigquery-gettingstarted.md`

[5] The ingested logs are: Google Argon (2017-2021), Google Xenon (2018-2022), Google Aviator, Google Icarus, Google Pilot, Google Rocketeer, Google Skydiver, Google Submariner, Google Daedalus, Cloudflare Nimbus (2017-2021), DigiCert Log Server, DigiCert Log Server 2, Symantec Log, Symantec Vega, Symantec Sirius, Certly.IO Log, Izenpe Log, WoSign Log, Venafi Gen2 CT Log, CNNIC CT Log, StartCom Log, PuChuangSiDa CT Log, Comodo 'Sabre' CT log, Comodo 'Mammoth' CT Log, and Up In The Air 'Behind the Sofa' Log.

[6] `https://productforums.google.com/forum/#!forum/chrome`

[7] Chrome does not evaluate CT compliance for non-encrypted requests, requests that are served from the HTTP disk cache, and connections for which the HTTPS certificate fails to validate (e.g. an expired or self-signed certificate) or chains to a locally installed root. We disregard such connections and requests because they would not be affected by broadening CT enforcement, so they are not of interest to our study.

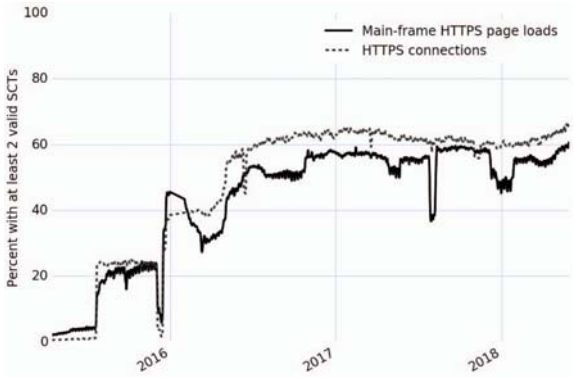

Fig. 4. The percentage of main-frame HTTPS page loads and HTTPS connections with at least two valid SCTs, as measured from Chrome from April 2015 through May 2018.

B. Understanding CT adoption from server support

We examine server scans to see which websites have adopted CT and why. Table I shows the percentage of sites that respond over HTTPS with a valid certificate and the percentage of those that are CT-compliant, as measured from January 2018 web crawler records. A small percentage serve SCTs but do not comply with the Chrome CT policy.

CT adoption is approximately even across the Alexa Top 10,000. We performed a logistic regression to predict CT compliance from Alexa rank among the Alexa Top 10,000 that support HTTPS, and a Wald test found no evidence that rank is a predictor ($p = 0.43$). However, CT adoption does lag in the long tail, as can be seen in Table I.

1) Reasons for adoption: In our dataset, most sites that have adopted CT appear to have done so because Chrome required them to, or because they are served by a major ecosystem player that is invested in CT's success. Table II shows the characteristics of CT-compliant websites among the websites we considered. Each table cell shows the percent of CT-compliant websites that may have adopted CT for one of the following reasons.

- **EV**: The website uses an EV certificate and therefore must support CT to receive Chrome's EV UI treatment.
- **Symantec**: The website uses a certificate issued by a Symantec-owned CA. Chrome requires such certificates to be logged if issued after June 1, 2016.
- **Cloudflare**: Most websites served by Cloudflare are not required to be logged by any browser, but Cloudflare logs all certificates that it serves nevertheless. This category in the table contains certificates obtained by Cloudflare for their hosted websites, which we identified by the presence of a *.cloudflaressl.com subject name or "CloudFlare, Inc." issuer name in the certificate. This excludes Cloudflare-hosted websites for which the customer provides their own certificate; Cloudflare logs and serves SCTs for these certificates as well.
- **GlobalSign**: In fall 2017, the GlobalSign CA began embedding SCTs in all certificates, motivated by Chrome's upcoming CT enforcement for all new certificates [25].

There may be some overlap among these categories: for example, a website can have a Symantec-issued EV certificate.

Browser enforcement (the EV, Symantec, and GlobalSign categories) and CDN support (Cloudflare) therefore drive a large portion of CT adoption. In Section V-A, we analyze the websites that are left behind with buggy or absent CT support despite browser requirements, usually due to bugs or misconfigurations in their CA's implementation of CT.

TABLE I
SERVER SUPPORT FOR CT.

	HTTPS	CT-compliant (out of HTTPS)	Serves noncompliant SCTs (out of HTTPS)
Alexa Top 10k	73.0%	54.0%	0.2%
CrUX Report	39.3%	36.0%	0.1%
HTTP Archive	54.8%	43.2%	0.1%

TABLE II
REASONS WHY CT-COMPLIANT WEBSITES MAY HAVE ADOPTED CT.

	EV	Symantec	Cloudflare	GlobalSign
Alexa Top 10k	13.4%	26.8%	42.4%	10.3%
CrUX Report	15.8%	37.7%	26.6%	12.3%
HTTP Archive	15.9%	30.8%	42.5%	9.4%

V. COMPLIANCE

How successfully do websites comply with Chrome's CT policy when Chrome requires them to comply? When they fail to comply, what is the cause? Noncompliance negatively affects users because it can lead to broken pages or warnings.

A website might fail to comply with a browser's CT policy due to developer choice or CA bugs. Some CAs and certificate owners choose not to log certificates because they don't want the certificate information to be public. Another possible cause of noncompliance is a bug or misconfiguration in the code that implements CT support. Such a problem can occur at a CA (when a CA is embedding SCTs into a certificate) or at a web server (for servers that implement their own CT support). Similar problems have arisen in the past when browsers rolled out new certificate requirements [26], [27].

We quantify two consequences of CT compliance failures that can occur in Chrome: (1) blocked connections, where a website or resource is blocked from loading due to CT noncompliance, and (2) EV downgrades, where a website loses its EV certificate UI due to CT noncompliance. In the week ending February 1, 2018, 15% of HTTPS connections were expected to be CT-compliant, meaning that either the connection would be blocked or its EV status would be removed if noncompliant. We find that the rate of noncompliance among these connections is very low, even historically when new CT enforcement requirements have rolled out.

A. Blocked connections

When a website is required to comply with the CT policy but doesn't, Chrome fails the connection, resulting in a certificate warning (Fig. 1) or a subresource silently failing. At the

time of writing, a blocked connection could occur because (a) Chrome requires CT compliance for the website's CA, or (b) the website has opted into CT via the `Expect-CT` header.

1) Current impact: CT noncompliance currently imposes a negligible end user impact. We consider Chrome metrics for the week ending February 1, 2018. Of all HTTPS connections and requests for which CT compliance was required, 100.0% were successfully compliant. (Note that multiple HTTP requests can share a single HTTPS connection.) CT compliance failures represented 0.02% of all main-frame certificate errors (that is, certificate errors that occur when loading the main resource of a page). These CT-related main-frame errors therefore make up a negligible fraction of all connections for which CT compliance is required.

From late July 2018 onwards, Chrome began additionally blocking connections when a website uses a noncompliant certificate issued after April 2018 (Section II-C2). In the week ending September 2, 2018, CT-compliance was required for 42.6% of connections, meaning that these connections would be blocked if they were noncompliant. Of these connections that were required to be CT-compliant, 99.5% were successfully compliant. Similarly, of HTTP requests for which CT was required, 99.7% were compliant. CT compliance failures represented 3.4% of all main-frame certificate errors during this time.

2) Historical impact: Historically, even when new CT requirements have gone into effect, the immediate user impact is small. For example, in fall 2016, Chrome began requiring CT compliance for certificates issued by the Symantec CA [14]. In the week ending September 9, 2016, shortly after the enforcement change was released to the Stable channel, CT errors represented only 1.2% of all certificate errors. During this time period, several popular websites were serving certificates with malformed SCTs that Chrome rejected. This mishap was due to an option provided by the CA to strip full domains names from the certificates submitted to CT logs. The resulting SCTs were not accepted by Chrome as valid. This option was intended for private internal domains, but it was selected accidentally by a number of large public websites [28].

Most instances of this error were quickly corrected: CT errors had dropped to 0.3% of all certificate errors by the week ending October 14, 2016.

3) Causes of blocked connections: CT-related breakage is fairly rare, but when it does occur, it is most commonly due to the name-stripping option discussed above, a CA implementation error, or a lack of CA support for CT.

We examined CT-caused certificate errors from the certificate error reports described in Section III-B. We used reports for CT errors that occurred during the week ending July 2, 2018 for Chrome 67 and the week ending September 2, 2018 for Chrome 68, the current Stable releases during these weeks, respectively.

The Chrome 67 dataset contained CT errors from 119 unique websites, and the Chrome 68 dataset contained 9,649. We manually inspected the 10 websites that caused the most CT errors during this week in each dataset. For privacy reasons, we restricted this analysis to websites that had been visited by Googlebot. We found the following four causes of noncompliance, with the breakdown shown in Table III.

TABLE III
THE CAUSES OF NONCOMPLIANCE AMONG THE TOP 10 WEBSITES THAT CAUSE CT ERRORS IN CHROME, FOR THE WEEKS ENDING JULY 2, 2018 (CHROME 67) AND SEPTEMBER 2, 2018 (CHROME 68).

	Name stripped	Not enough SCTs	Not diverse SCTs	No SCTs
Chrome 67	8	1	1	0
Chrome 68	0	0	0	10

- **Name stripping**. These websites serve malformed SCTs as part of the name stripping option described above.
- **Not enough SCTs**. These websites serve certificates with embedded SCTs, but not enough of them as required by the Chrome CT policy. (The policy requires a varying number of embedded SCTs depending on the certificate lifetime.) These errors likely represent a bug in the CA's implementation of CT.
- **Not diverse SCTs**. These websites serve certificates with embedded SCTs, but they do not satisfy the log diversity requirement of the Chrome CT policy (for example, the SCTs come from all Google logs). Again, these errors likely represent a bug in the CA's implementation of CT.
- **No SCTs**. These websites do not serve SCTs at all, indicating that their CAs have not yet implemented CT support despite Chrome's requirement.

B. EV downgrades

If an EV certificate is not compliant with Chrome's CT policy, Chrome removes the EV browser UI but does not block the connection. This restriction went into effect in early 2015. At the time of rollout, Chrome included a whitelist of EV certificates that had appeared in CT logs as of January 1, 2015 but did not have embedded SCTs. Certificates on this whitelist were treated as CT-compliant until June 2017, when the whitelist was little-used enough that it was removed.

Chrome does not support EV on mobile platforms, so this section only includes data from desktop Chrome.

1) Current impact: Currently, EV downgrades are rare but not unheard-of. In the week ending February 1, 2018, EV status was removed for 1.0% of connections with EV certificates. 90% of these removals were due to an insufficient number of SCTs, and the remaining 10% failed to meet the log diversity requirement.

2) Historical impact: Since the EV CT requirement came into effect, the frequency of EV downgrades has always been low, as shown in Fig. 5. EV downgrades never exceeded 4% of connections with EV certificates. This result is largely thanks to the EV whitelist, which preserved the EV UI on more than 70% of connections initially and still accounted for about 7% of EV connections a year later.

The Chrome EV plan was announced in September 2013 [29], but most CAs did not begin embedding SCTs

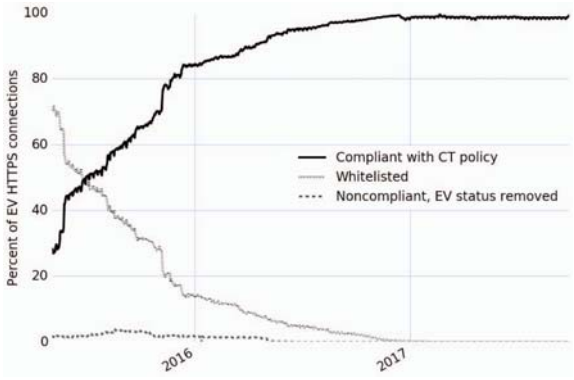

Fig. 5. The CT-compliance of HTTPS connections with Extended Validation certificates, as measured from Chrome from May 2015 through October 2017 (approximately the period during which the EV whitelist was in effect).

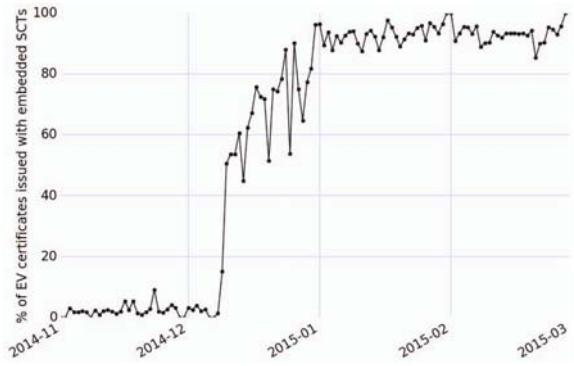

Fig. 6. The percentage of EV certificates issued each day with embedded SCTs, as observed in CT logs. After January 1, 2015, Chrome required all EV certificates to be logged.

into EV certificates until the month before the January 2015 deadline. Fig. 6 shows the percentage of EV certificates issued each day with embedded SCTs, as observed in CT logs.

Some CAs continued to issue EV certificates after January 2015 without embedded SCTs, even though these certificates would not receive EV UI treatment unless the site owner served SCTs themselves via TLS extension or OCSP response. Table IV shows the count of EV certificates issued after January 1, 2015 without embedded SCTs, grouped by issuing organization. Some organizations are combined because they represent the same corporate entity. (For example, the issuer organization names "GeoTrust Inc.", "VeriSign, Inc.", and "Symantec Corporation" were all Symantec brands during this time period, so they are all combined under "Symantec Corporation".) The table shows the 10 organizations that have issued the most EV certificates without embedded SCTs.

Fortunately, the end user impact of these certificates has been low, perhaps because they are used on less popular websites or on websites that are not accessed by Chrome users. We also note that prior work suggests that users may not even notice the absence of an EV indicator [30], [31].

TABLE IV
TOP ISSUERS OF EV CERTIFICATES WITHOUT EMBEDDED SCTS FROM JANUARY 1, 2015 THROUGH JULY 6, 2018, AS OBSERVED BY CT LOGS.

Issuing organization	EV certificates w/o SCTs	Total EV certificates	% w/o SCTs
Verizon Cybertrust Security	8550	8556	99.9%
Symantec Corporation	1923	495528	3.9%
SwissSign AG	1719	1908	90.1%
Certplus	1391	1391	100.0%
Cybertrust Japan Co., Ltd	1373	24748	5.5%
StartCom CA	443	7356	6.0%
D-Trust GmbH	245	725	33.8%
GlobalSign nv-sa	149	52675	0.3%
QuoVadis Limited	142	24398	0.6%
DigiCert Inc	125	210322	0.1%

VI. USER IMPACT

In this section, we measure the user impact of CT in two key areas: the performance costs of SCT validation and the user experience when CT causes errors or breakage.

A. Performance

SCT validation in Chrome. CT was designed to minimize performance impact by allowing clients to verify public key signatures from logs rather than blocking connections while communicating with the logs directly. In Chrome 69, SCT validation has a moderate performance impact compared to certificate validation.

For the week ending September 24, 2018, among HTTPS connections with SCTs, 99% of SCT validations complete in under 13.3 milliseconds, with a mean of 1.9 and median of 1.1 milliseconds. In comparison, certificate validations take 88 milliseconds on average, with a median of 14 milliseconds.

Chrome caches the results of certificate validations, so the actual time spent validating certificates when setting up an HTTPS connection is 18 milliseconds on average, with a median of less than 1 millisecond. A similar optimization could be applied to SCT validation to reduce the performance impact of CT. In this design, Chrome would cache the results of SCT validations and would not repeat validation for the same set of certificate and SCTs when it has a result cached already.

Performance issues in Firefox. Firefox's experimental CT implementation has been blocked for over a year on an unresolved performance regression [32]. The underlying cause is not yet clear, and significant investigation and investment is needed before Firefox's CT code will be enabled by default [33]. This issue demonstrates that CT deployment is a substantial investment for browser implementers, a hurdle that we discuss further in Section IX-B.

B. User Experience of Errors

Though users encounter CT errors rarely, they react to them in unsafe ways. These behaviors are especially concerning because CT errors tend to occur on security-sensitive sites.

1) Error clickthrough: Users bypass CT errors, choosing to proceed in spite of the security warning, at a much higher rate than certificate errors in general. We note that the error UI is nearly identical across all types of certificate errors, and that prior work has been successful at improving the UI's adherence but not comprehension [34].

In the week ending July 2, 2018, users proceeded through 47.8% of CT errors shown in Chrome 67, compared to 28.0% of all bypassable certificate errors[8]. (For comparison, users proceeded through 4.9% of Safe Browsing errors that warn about phishing or malware.)

Similarly, in Chrome 68, which began enforcing CT for newly issued certificates. During the week ending September 2, 2018, when Chrome 68 was the current Stable release, users proceeded through 49.0% of CT errors, compared to 28.2% of all bypassable certificate errors.

This finding underscores the need to deploy CT gradually with minimal breakage. Otherwise, users might learn to bypass certificate errors whenever they see them, undermining the security benefits of CT and of HTTPS in general.

2) Types of affected sites: Users experience CT errors most often on websites in important, security-sensitive domains, such as government portals and financial services. We manually categorized the top 10 websites that caused CT errors in Chrome from Table III. For the Chrome 67 websites, four were financial services, two were enterprise administrative portals, and one was a government site. Three lacked sufficient information for us to be able to categorize them. Of the Chrome 68 websites, eight were government sites, one was a corporate login portal, and one could not be categorized. The Chrome 68 results are consistent with prior findings that government websites are disproportionately responsible for certificate errors in Chrome [18].

3) Help forum case studies: To further understand how CT errors impact end users, we reviewed CT-related posts in the Chrome product help forum as described in Section III-E. The Appendix contains the full results of our coding procedure.

Many of the help forum posts came from a Chrome bug during fall 2016. Out-of-date Chrome builds exhibited this bug by showing CT errors for large numbers of websites that were correctly CT-logged [35]. This incident, while not a CT compliance issue, provides insight into how users react to widespread CT errors, and also demonstrates the complexity of building web browser support for CT (Section IX-B).

Incorrect solutions. 60% of threads were labelled as "Incorrect fix or explanation suggested or tried, or issue was perceived to be solved by something unrelated". (For comparison, CT was correctly identified as the source of the issue in 41% of the threads.) Users commonly disabled antivirus software, checked system date settings, or disabled extensions in an attempt to fix the error.

> *I have tried resetting to default settings (so disabling all extensions).* [36]

[8]The Chrome error UI does not allow users to bypass certificate erorrs on sites with certain security settings, such as HTTP Strict Transport Security.

In these scenarios, users lose their data and settings without resolving the issue. Users might also end up in a less secure state, for example by disabling extensions that make their browsing more secure.

Bypassing errors. Users bypassed CT errors by clicking through the warning or by trying different browsers. In 8% of the threads, users mentioned that they clicked through the warning or tried to, and in 19% of the threads, users noted that affected websites worked in other browsers, or that they would even consistently use another browser to access the affected sites:

> *I had to download another browser, which im starting to like.* [36]

These responses show that browsers take a first-mover risk by implementing new security requirements such as CT: breakage could cause users to switch to other browsers. The tendency to switch browsers also demonstrates that other browsers must adopt CT in order for CT to have its intended security benefits. Otherwise, in an attack that uses a noncompliant certificate, users might simply ignore the warning and switch to another browser that leaves them vulnerable.

Poor user experience. In 9% of threads, users expressed anger, frustration, or impatience, and in 9%, they mentioned a specific task (such as homework or bill-paying) that they could not complete because of the error:

> *This makes it nearly impossible to manage bills.* [37]

These threads demonstrate the generally poor user experience when users encounter frequent warnings on the web.

VII. Risks

CT enforcement has not imposed widespread breakage on end users so far, as discussed in Section V. However, the CT ecosystem is young and end users could still be affected as CT enforcement rolls out more broadly. In this section, we present data and case studies to examine two risks of CT enforcement: (1) log disqualification or distrust, which might require website owners or CAs to take action in order to avoid breakage for end users, and (2) server-side SCT serving, which can cause disruption if implemented improperly. Our aim is to assess the size of the breakage that these risks might impose on end users and uncover best practices for avoiding them.

A. Log distrust

Chrome requires logs to meet a set of requirements, such as maintaining a certain uptime, incorporating certificates into the log quickly, and always presenting consistent views of the log [38]. When a log fails to meet these requirements, it might be removed from Chrome's list of logs and Chrome may stop accepting its SCTs. As a result, websites using the log's SCTs might no longer comply with Chrome's CT policy, leading to warnings or breakage for end users. Chrome responds to different types of log failures with different actions:

- For an unrecoverable security failure, such as a compromise of a log's private key, a log can be *fully distrusted*. None of the log's SCTs will be accepted as valid anymore. This is the type of distrust that causes the most

breakage for end users because it can break existing certificates; fortunately, so far it has occurred only once, for a virtually unused log [39].

- When a log has a single security incident, it can be *disqualified*. A certificate with an embedded SCT from the disqualified log will continue to be accepted as valid as long as the certificate has SCTs from other logs that have not been distrusted or disqualified [12]. However, SCTs delivered via TLS extension or OCSP response must be replaced, and Chrome will not accept new SCTs from the log. Log disqualification can result in disruption for end users – for example, if a server fails to update the SCTs that it is serving in the TLS extension. However, a log disqualification is usually far less disruptive than full distrust because it usually does not require many existing certificates to be replaced.
- When a log has an operational incident that does not pose a significant security risk, the log can be *frozen*. Existing SCTs from the log continue to be accepted exactly as before, but the log cannot issue new SCTs. Log freezing is usually done with the cooperation of the log operator and poses minimal risk of negative impact to end users.

We focus on disqualification when discussing log distrust because it has happened often historically and can cause user disruption. In contrast, log freezing happens often but does not affect end users, and full distrust is nearly unheard-of so far.

1) Measuring disqualification risk: A small number of CT logs carry a disproportionate risk of causing negative impact to end users.

When a log is disqualified, there are two effects: (1) websites serving the log's SCTs in the TLS extension or OCSP response must replace them, and (2) existing certificates might need to be replaced, if multiple SCTs embedded in the certificate come from logs that have been disqualified.

In practice, (2) is the larger risk to end users because it requires site owners to take action even though they might not have any knowledge of CT, having simply acquired a certificate from a CA that embeds SCTs. Historically, users encounter frequent certificate errors when browsers make changes that require site owners to replace certificates. For example, deprecated SHA-1 certificates accounted for 9.4% of all certificate errors in Chrome in the two months after Chrome began showing full-page certificate warnings for them [18].

We find that a single pair of logs is disproportionately "load-bearing": if both of the logs were disqualified, a large number of sites' existing certificates would no longer be CT-compliant. Table V shows the number of websites in the Alexa Top 10,000 with certificates that would no longer be accepted as valid if a pair of logs were to be disqualified. The "Sites with affected certs" column shows the number of sites with SCTs embedded in certificates which would become noncompliant if the given logs were disqualified. These sites would need to either update their certificates or begin serving compliant SCTs via TLS extension or OCSP response. If the Google Pilot and Symantec CT logs were both disqualified, 458 websites – 12% of the CT-compliant websites in the Alexa list – would no longer be serving CT-compliant certificates.

TABLE V
THE RISK OF LOG DISQUALIFICATIONS FOR THE ALEXA TOP 10,000 AS OF FEBRUARY 2018. THE TABLE OMITS LOGS OR PAIRS OF LOGS THAT WOULD NOT IMMEDIATELY AFFECT ANY EXISTING CERTIFICATES IF DISQUALIFIED.

Disqualified logs	Sites with affected certs
Google Pilot, Symantec CT	458
Google Pilot, Digicert CT2	23
Google Rocketeer, Symantec CT	18
Google Pilot, Digicert CT1	17
Google Skydiver, Digicert CT2	14
Google Rocketeer, Digicert CT2	2
Google Skydiver, Symantec CT	1
Google Pilot, Symantec Vega	1

Note that Table V simulates log disqualifications as of February 1, 2018. Logs can also be retroactively disqualified, which would affect certificates that had been issued since the retroactive disqualification date [40]. Depending on the effective disqualification date, retroactive disqualifications can have an even bigger impact on end users, because newly-issued certificates might need to be replaced.

2) Disqualification incidents: To understand how log disqualification happens and can be avoided, we describe several anecdotes about log failures and how they were addressed.

Unrecoverable incidents. In February 2017, the Venafi log presented two inconsistent views, violating the append-only property of CT logs [41]. The incident was caused by an Amazon Web Services outage, in which the log published a view based on a backup out of sync with the log's current set of certificates. The log was retroactively disqualified from the point at which it had published an inconsistent view [40].

In practice, this disqualification appeared to have little negative impact on end users. The disqualification was estimated to affect fewer than 2,000 certificates [40]. CT accounted for 0.05% of all certificate errors in Chrome in the week ending June 15, 2017, shortly after the disqualification took effect in Chrome's Stable channel.

Vigilant monitoring and speedy reporting helped minimize the impact on end users. The incident was noticed by CT monitoring software written and operated by an external researcher. Had the inconsistent views not been noticed and acted on quickly, they might have been uncovered much later on. A retroactive disqualification might have then affected many more certificates that had been issued in the meantime, possibly causing greater disruption for end users if those certificates were not replaced by site owners.

Recoverable incidents. At the time of writing, Cloudflare is in the process of bringing up a new set of CT logs which have recovered from two incidents of note.

In one incident, the log presented two subsequent, consistent views, but failed to produce the full set of certificates between the two views [42]. Further investigation revealed that the published views were calculated using a set of certificates which was different than the consensus view on which the

underlying datastore eventually settled. Fortunately, a redundant datastore existed, allowing the log operators to reconstruct the certificates corresponding to the published views. While the main datastore (Kafka) did not allow the log operators to reorder data, they modified their log implementation to specially handle the problematic certificates. This fix reconciled the situation, allowing the log to serve certificates matching the published views. Since the published views were never inconsistent with each other, the incident was not treated as an unrecoverable failure.

In another pre-production incident, the log operators discovered an implementation bug that resulted in publishing inaccurate views of the log. To prevent a similar situation from occurring in production, they produced two independent implementations and now check the results against each other before publishing them. This strategy helps prevent the logs from publishing inconsistent or incorrect views.

Neither of these incidents affected users, because they occurred during development or Chrome's monitoring period before accepting new logs [38]. The monitoring period is useful not just for ensuring that logs uphold their security properties but also for minimizing user-visible breakage, by helping log operators discover and fix operational problems.

B. Server-side SCT delivery

Some websites choose to serve SCTs themselves in the TLS extension, rather than having their CAs provide SCTs by embedding them in their certificates or serving them in OCSP responses. A common reason to use the TLS extension is to avoid sending SCTs to clients that don't signal support for them in the TLS handshake: it wastes bandwidth to send SCTs to clients that don't use them.

In some ways, the TLS extension can avoid negative impact on end users. A site owner can serve SCTs via TLS extension to react quickly to a log distrust event, rather than waiting for their CA to provide a new certificate with valid SCTs.

However, if implemented improperly, server-side SCT delivery can pose risks to end users. For example, a server must use up-to-date log metadata when obtaining its SCTs; otherwise, it could serve an SCT from a log that has been distrusted, causing users to see warnings or breakage on that website.

In this section, we measure the prevalence of server-side SCT delivery and present an anecdote about SCT delivery gone wrong.

1) Measuring server-side SCT delivery: Server-side SCT delivery tends to be implemented by a small number of large organizations, such as performance- and bandwidth-conscious CDNs. The TLS extension accounts for a disproportionately large percentage of SCTs on the web, despite the fact that most individual site owners do not choose to serve SCTs themselves.

In the week ending February 1, 2018, 47.97% of SCTs observed in Chrome came from the TLS extension. (0.01% came from OCSP responses and 52.02% were embedded in certificates.) 50% of CT-compliant websites in the Alexa Top 10,000 serve SCTs in the TLS extension. This compares to 30% in the Chrome User Experience Report and 44% in the HTTP Archive. Prior work has also found that more popular sites tend to serve SCTs themselves [43].

It is therefore important that implementations of SCT delivery solidify around best practices. Problems in server-side SCT delivery could affect end users at a large scale.

2) Server-side SCT delivery incidents: In mid-2017, a company that owns a large set of websites briefly served an SCT via TLS extension that many Chrome clients did not accept, accounting for the mid-2017 dip in valid SCTs in Fig. 4.

The incident occurred because the websites began using a log that was trusted in the latest version of Chrome, but that version had not yet been widely deployed to users yet. For users with slightly out-of-date Chrome versions, the websites appeared to be serving SCTs from an unknown log. Fortunately, CT was not yet required for these websites and the incident was not visible to end users.

This occurrence underscores the importance of adopting CT gradually before it is fully required, so that problems can be discovered and fixed before incurring negative user impact.

We note that commodity implementations of server-side TLS delivery do not yet exist. Implementations for servers such as nginx and Apache are experimental, not well-tested, and not solidified around best practices [44], [45]. Developing mature implementations of server-side SCT delivery is therefore an important area of future work.

VIII. Discussion: Design Principles

Our measurements show that CT has been adopted with minimal breakage so far. In this section, we discuss three CT design properties that have made this possible.

A. Small number of first-movers

The burden of CT adoption currently falls largely on CAs. Individual websites get the security benefits of CT without having to take action. For example, when Chrome began requiring CT for EV certificates, CAs began embedding SCTs into EV certificates so that the certificates would continue to trigger the EV UI. If EV CT enforcement had required individual site owners to take action, it is unlikely that the noncompliance rate would have been as low as shown in Fig. 5. Website owners can take action to improve the security properties that they get from CT (for example, by deploying the `Expect-CT` header), but by design they do not need to.

In contrast, we consider a web PKI technology that depends on individual site owners to adopt: HTTP Public Key Pinning (HPKP). HPKP allows sites to "pin" their connections to individual public keys, mitigating the site's vulnerability to a CA compromise [46]. HPKP is notoriously difficult for site owners to implement correctly [47]–[49]. Perhaps as a result, HPKP has seen very low adoption across the web [50], contributing to Chrome's recent decision to deprecate HPKP [51]. While it can be tricky to adopt CT correctly, one CA or CDN can do the work to protect all of its customers, leading to high adoption without the involvement of individual site owners.

B. Staged enforcement

By rolling out CT enforcement in stages, ecosystem problems are flushed out early and fixed before they have a bigger negative impact on end users. Moreover, each stage of enforcement carries a smaller risk of breakage than if enforcement were to be turned on universally at once.

For example, consider the incident discussed in Section V-A, where a number of websites stopped working when Chrome rolled out CT enforcement for a large CA. This mishap imposed some negative impact on end users, accounting for 1.2% of certificate warnings in Chrome, but the impact could have been much larger if multiple CAs had been subject to enforcement at once.

To limit the impact of each stage, enforcement requirements can be constrained by several strategies. For EV certificates, Chrome shipped a whitelist of certificates that had been logged but were not served with SCTs. This strategy accomplishes the goal of the enforcement requirement – ensuring that all relevant certificates are logged – and minimized the impact of on end users. In July 2018, Chrome began requiring CT for all certificates issued after April 2018. An issuance date cut-off compromises on security because it does not require old or backdated certificates to be logged, but it allows Chrome to require CT for a broader class of certificates, for which a whitelist would not be feasible, without widespread breakage.

C. Browser policy

Chrome's CT policy is carefully chosen to mitigate the risks of the early CT ecosystem. For example, the policy allows logs to be frozen or disqualified rather than fully distrusted in response to certain types of log failures (Section VII-A). In recent months, freezing and disqualification have been deemed appropriate responses for several operational and security incidents [52], [53]. This aspect of the policy benefits users because logs can be frozen or disqualified with less disruption than a full distrust event.

IX. Discussion: Deployment Hurdles

In this section, we discuss aspects of CT that might present roadblocks for future similar systems.

A. Sponsorship from major players

Successful CT deployment has relied heavily on sponsorship and investment from major ecosystem players such as Google and Cloudflare. For example, as discussed in Section V-B, EV certificates were not widely logged until Chrome began dropping the EV UI for noncompliant certificates. Moreover, Cloudflare has driven a substantial portion of CT adoption (Table II). It is unclear that CT would have been widely adopted without the investment of these large organizations. Future similar systems may not see successful deployment without similar investment from major ecosystem players.

B. Large implementer investment

Part of CT's success has been due to the fact that it does not require individual site operators to take action (Section VIII-A), but one of its challenges is that it requires substantial implementation investment from other entities, including browser vendors.

Browsers face a first-mover risk when considering whether to implement new security requirements like CT. As discussed in Section VI-B3, users might simply switch to another browser in the face of an attack or misconfiguration.

Initial implementation can be a substantial engineering effort. For example, Firefox's experimental CT implementation has been disabled by default for over a year due to unresolved regressions [33]. Firefox developers have expressed concerns about whether the security properties of CT are valuable enough to warrant this investment [54].

Finally, CT presents an ongoing maintenance burden, and when bugs are introduced, they can be costly. This is illustrated by the Chrome bug discussed in Section VI-B3, in which out-of-date builds displayed spurious CT warnings, frustrating users and in some cases inducing them to switch browsers.

X. Open problems

Even after all browsers require SCTs for all websites, other parts of CT need to be deployed to achieve the system's full security goals. Below we discuss open CT research areas and how they might impact end users.

A. Log auditing

CT defines protocols for auditing logs and verifying that they are behaving correctly. Experimental implementations of these technologies exist but are not yet widely deployed [55], [56]. Widespread log auditing may yet reveal two sources of disruption to end users. First, it might uncover log misbehavior that is currently going unnoticed, leading to more log disqualifications. Indeed, early implementations have already noticed misbehavior that resulted in log disqualifications [57], [58]. Second, log auditing protocols pose interesting questions about the privacy of end user browsing habits [59]. Once such protocols are fully specified and deployed, there is more work to be done to understand the actual impact on users' privacy and how it matches users' security and privacy expectations.

B. Name redaction

Various organizations have expressed the desire to be able to log certificates while redacting the full domain name from the logged certificate [60]. Some domain names, such as internal corporate hostnames, are understandably secret, but these hostnames can leak in various other ways (for example, DNS queries and domain registrations). Moreover, the goal of CT – full transparency into the issuance of public certificates – is compromised if browsers accept SCTs for redacted certificates. The community has not yet settled on whether it is desirable to allow information to be redacted from logged certificates.

As CT enforcement rolls out more broadly, websites might choose to not log their certificates due to lack of redaction

support in browsers, resulting in warnings for end users. Future work on name redaction can elucidate this risk by examining redaction use cases, analyzing the security properties of the current proposals, and exploring alternatives.

C. User reaction to CT errors

Chrome currently displays CT errors with a UI that is similar to other certificate validation errors, such as an expired certificate. In Section VI-B, we explore how users react to these errors by analyzing clickthrough rate and Chrome help forum posts, but future work can explore users' reactions to CT errors in more depth. For example, why do users proceed through CT errors at a higher rate than other certificate errors, and is there a better UI for communicating CT errors to users? Answers to these questions can help inform the warning design in Chrome and other browsers as they adopt CT.

XI. Related work

A. Browser warnings

Warnings science. A long line of work establishes the harm of frequent warnings, which motivates our desire to minimize warnings introduced by CT enforcement. Böhme et al. argue that false alarms consume from a user's finite lump-of-attention [2]. The negative effect of frequent warnings and prompts has been studied extensively in the context of permissions systems [61], [62], web browser warnings [3], [63], and popup dialogs like installation prompts [64], [65].

Causes of HTTPS errors. Several studies have examined the causes of HTTPS certificate errors. Acer et al. collected reports from browser warnings to analyze the top causes of false-positive certificate errors in the wild [18]. This study did not analyze CT errors specifically, perhaps because CT errors are rare relative to other types of certificate errors. Previous work, predating the widespread deployment of CT, studied the underlying causes of certificate errors from passive network observations [66] and by surveying developers [67].

B. Web PKI measurement

CT measurement. Perhaps most related to our work are previous studies of the CT ecosystem. Amann et al. combined passive and active scans to measure the adoption of several new web PKI technologies, including CT [43]. Like us, they conclude that CT has gained significant momentum and adoption, largely thanks to the investments of large ecosystem players like Google. We observe a higher percentage of CT-compliant traffic than in their study, likely due to differences in vantage point or time (Fig. 4). We also corroborate their finding that Symantec's name stripping option, described in Section V-A, is a common source of invalid SCTs.

In contrast to Amann et al.'s study, we provide measurements focused on the user experience of CT, particularly related to errors and breakage. We specifically measure how many connections are required to be CT-compliant but aren't. We also report CT compliance against the exact Chrome CT policy, whereas Amman et al. use a simplified version of the policy, only checking whether a connection contains one Google and one non-Google SCT. The nuances of the complete policy account for some of the top noncompliant sites that cause CT errors in Chrome (Section V-A).

VanderSloot et al. focus on building a complete view of publicly issued certificates by collecting certificates from CT logs as well as other sources [68]. Their work assesses how well CT represents the corpus of certificates used on the public internet, whereas our study measures the adoption of CT on the web and the breakage that it imposes on users.

Gustafsson et al. characterize the differences in size, growth rate, and certificate acceptance policies among various CT logs [69]. This work is primarily concerned with characterizing the certificate ecosystem from the logs' perspective, and also provides a view of log coverage of popular domains based on a one-week passive observation of a university network. In contrast, our work is concerned with how much web traffic successfully serves SCTs (as opposed to how many websites use certificates that appear in CT logs).

HTTPS measurement. Our paper builds on work that measures the HTTPS ecosystem more generally. Felt et al. propose a variety of metrics and perspectives for measuring HTTPS adoption [50]. Internet-wide scans and passive measurements are used in several other works to identify trends in certificate issuance and HTTPS adoption [70]–[73]. Our work adapts these methods for measuring CT adoption: for example, we consider both HTTP request and page load metrics for CT compliance, as Felt et al. do for HTTPS adoption [50].

C. Building on CT

In Section X, we identify several areas of future work that can help minimize the negative end user impact of CT. Early work has begun to explore these areas, particularly in protocols to uncover log misbehavior. This work includes gossip protocols [74] and privacy-preserving schemes that allow clients and other parties to discover and report misbehavior [75].

XII. Conclusion

Certificate Transparency can improve the security of the web PKI, but end users will suffer if they are inundated with warnings and breakage as CT enforcement rolls out. In this paper, we showed that Certificate Transparency has been adopted across a significant fraction of the web with minimal breakage for end users. When users do encounter errors or warnings, however, they react unsafely, so it is critical to CT's success that breakage rates stay low. We further measured and studied risks in CT deployment that could cause disruption for end users in future: namely, the risks of log disqualifications and improperly implemented server-side SCT delivery. We observed that CT's success so far can be attributed to a number of design properties: for example, CT can be deployed by a small number of first-movers, and widespread adoption does not require individual action by site owners. Finally, we identified hurdles in the deployment of CT that might present challenges to similar systems in future.

TABLE VI
THE CODEBOOK USED TO LABEL CHROME HELP FORUM THREADS FOR CT ERRORS, AND THE NUMBER OF THREADS ASSIGNED EACH LABEL BY BOTH RATERS. 75 THREADS WERE CODED IN TOTAL.

Label	Description and/or examples	# threads
Incorrect fix or explanation suggested or tried, or issue was perceived to be solved by something unrelated	"the malwarebytes workt great" "... go to chrome://extensions and uncheck Enabled for each extension one by one."	45
Recommended or tried updating Chrome	This is the correct solution for the bug described in Section VI-B3 in which out-of-date Chrome builds showed spurious CT errors. "Please ensure you're using the latest version of Chrome."	33
Multiple sites affected	"ON EVERY WEBSITE I TRY TO VISIT"	31
CT identified as the problem	"Chrome now requires that all Symantec and Symantec associated groups (including GeoTrust and VeriSign) SSL security certificates be registered into Certificate Transparency"	31
Tried other browsers, or will stop using Chrome for some or all sites	"other browsers don't experience this issue" "i guess I have to stop use google chrome"	14
Recommended or tried changing release channels	"Can you try Chrome Beta and check if it helps."	12
New machine	"its a brand new chromebook and I cant use it!"	11
Thread is for non-CT error (or not clear if CT error)	The CT error code might show up incidentally in the thread, but not enough information is given to tell that the user had actually encountered a CT error.	8
Can't complete task	"This makes it nearly impossible to manage bills"	7
Anger, frustration, impatience	"I understand that Google wants to protect us from malicious sites, but offering the only solution as turning protection off is plain stupid. Deal with it please."	7
User tried to bypassing error, bypassing given as advice, or user notes that they can't bypass it	"But unlike some websites where you can continue anyway, I can't open Netflix at all"	6
Other user with different issue (e.g., different cert error)	"I've the same issue here... SHA-1 Certificate"	4
Request for technical details	"Highlight-and-copy... all the certificate blocks in the 'PEM encoded chain', and paste them into this thread..."	4
Site is trustworthy or reliable	"this site is TRUSTWORTHY" "I get the same NET ERR message from a very reliable site"	4
Webpage appears broken	Subresources are failing to load, e.g., "in some cases when I try to a load a page only the texts load"	4
Other support channel tried or suggested	"you may consider contacting the admin team to receive further help as the settings are managed by them"	4
Issue resolved by itself	"And as I was typing this, the issue is gone."	3
Apprehension about bypassing error	"I use to help me bank so I'm REALLY apprehensive about logging in"	2

APPENDIX
CHROME PRODUCT HELP FORUM DATA

Table VI shows the full set of labels that we applied to Chrome help forum threads about CT errors, and how many threads each label applied to (when both raters assigned the label to a thread).

REFERENCES

[1] "An update on attempted man-in-the-middle attacks," August 2011, https://security.googleblog.com/2011/08/update-on-attempted-man-in-middle.html.

[2] R. Böhme and J. Grossklags, "The security cost of cheap user interaction," in *Proceedings of the 2011 New Security Paradigms Workshop*, ser. NSPW '11. New York, NY, USA: ACM, 2011, pp. 67–82. [Online]. Available: http://doi.acm.org/10.1145/2073276.2073284

[3] J. Sunshine, S. Egelman, H. Almuhimedi, N. Atri, and L. F. Cranor, "Crying wolf: An empirical study of SSL warning effectiveness," in *Proceedings of the 18th Conference on USENIX Security Symposium*, ser. SSYM'09. Berkeley, CA, USA: USENIX Association, 2009, pp. 399–416. [Online]. Available: http://dl.acm.org/citation.cfm?id=1855768.1855793

[4] 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 *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*, ser. IMC '11. New York, NY, USA: ACM, 2011, pp. 427–444. [Online]. Available: http://doi.acm.org/10.1145/2068816.2068856

[5] J. Clark and P. C. van Oorschot, "SoK: SSL and HTTPS: revisiting past challenges and evaluating certificate trust model enhancements," in *2013 IEEE Symposium on Security and Privacy*, May 2013, pp. 511–525.

[6] "What is certificate transparency?" https://www.certificate-transparency.org/what-is-ct.

[7] Y. Pettersen, "The Transport Layer Security (TLS) Multiple Certificate Status Request Extension," June 2013, https://www.ietf.org/rfc/rfc6961.txt.

[8] B. Laurie, A. Langley, and E. Kasper, "Certificate transparency," June 2013, https://tools.ietf.org/html/rfc6962.

[9] "PKI:CT," December 2014, https://wiki.mozilla.org/PKI:CT.

[10] "Certificate transparency policy," 2018, https://support.apple.com/en-us/HT205280.

[11] T. Shinder, "Certificate transparency," April 2018, https://blogs.msdn.microsoft.com/azuresecurity/2018/04/25/certificate-transparency/.

[12] "Certificate Transparency in Chrome," May 2016, https://github.com/chromium/ct-policy/blob/21cb3623c005ae0118cbbd911e10e6b44eb28528/ct_policy.md.

[13] "Extended Validation SSL FAQ," https://www.digicert.com/extended-validation-ssl.htm.

[14] R. Sleevi, "Sustaining digital certificate security," October 2015, https://security.googleblog.com/2015/10/sustaining-digital-certificate-security.html.

[15] E. Stark, "Expect-ct extension for HTTP," August 2017, https://tools.ietf.org/html/draft-ietf-httpbis-expect-ct-02.

[16] "Certificate Transparency in Chrome - Change to Enforcement Date,"

April 2017, https://groups.google.com/a/chromium.org/d/msg/ct-policy/sz_3W_xKBNY/6jq2ghJXBAAJ.
[17] "Keynote (Chrome Dev Summit 2016)," https://www.youtube.com/watch?v=eI3B6x0fw9s.
[18] M. Acer, E. Stark, A. P. Felt, S. Fahl, R. Bhargava, B. Dev, M. Braithwaite, R. Sleevi, and P. Tabriz, "Where the wild warnings are: Root causes of Chrome certificate errors," 2017.
[19] "How are Alexa's traffic rankings determined?" https://support.alexa.com/hc/en-us/articles/200449744-How-are-Alexa-s-traffic-rankings-determined-.
[20] "Chrome User Experience Report," January 2018, https://developers.google.com/web/tools/chrome-user-experience-report/.
[21] "About the HTTP archive," http://httparchive.org/about.php.
[22] D. R. Thomas, "A general inductive approach for analyzing qualitative evaluation data," *American Journal of Evaluation*, vol. 27, no. 2, pp. 237–246, 2006. [Online]. Available: https://doi.org/10.1177/1098214005283748
[23] L. L. Kupper and K. b. Hafner, "On assessing interrater agreement for multiple attribute responses," *Biometrics*, vol. 45, no. 3, pp. 957–967, 1989. [Online]. Available: http://www.jstor.org/stable/2531695
[24] J. R. Landis and G. G. Koch, "The measurement of observer agreement for categorical data," *Biometrics*, vol. 33, no. 1, pp. 159–174, 1977. [Online]. Available: http://www.jstor.org/stable/2529310
[25] "Upcoming Baseline Requirement Changes," https://www.globalsign.com/en/ssl-information-center/baseline-requirements/.
[26] R. Andrews, "[cabfpub] Incident Report: SHA-1 Certificates issued after 31 December 2015," January 2016, https://cabforum.org/pipermail/public/2016-January/006519.html.
[27] S. Carletti, "Clarification about Ballot 193 and validity of a certificate issued on March 1st," April 2018, https://groups.google.com/d/msg/mozilla.dev.security.policy/-o2iN4GQbGY/KmErqpFDCAAJ.
[28] A. Ayer, "Why Chrome 53 is rejecting Chase Bank's Symantec certificate," September 2016, https://sslmate.com/blog/post/ct_redaction_in_chrome_53.
[29] R. Sleevi, "[cabfpub] Upcoming changes to Google Chrome's certificate handling," September 2013, https://cabforum.org/pipermail/public/2013-September/002233.html.
[30] C. Jackson, D. Simon, D. Tan, and A. Barth, "An evaluation of extended validation and picture-in-picture phishing attacks," January 2007. [Online]. Available: https://www.microsoft.com/en-us/research/publication/an-evaluation-of-extended-validation-and-picture-in-picture-phishing-attacks/
[31] S. E. Schechter, R. Dhamija, A. Ozment, and I. Fischer, "The emperor's new security indicators," in *2007 IEEE Symposium on Security and Privacy (SP '07)*, May 2007, pp. 51–65.
[32] "certificate transparency signature verifications negatively impact TLS handshake performance," 2017, https://bugzilla.mozilla.org/show_bug.cgi?id=1353216.
[33] "Re-enable Certificate Transparency telemetry collection," 2017, https://bugzilla.mozilla.org/show_bug.cgi?id=1355903.
[34] A. P. Felt, A. Ainslie, R. W. Reeder, S. Consolvo, S. Thyagaraja, A. Bettes, H. Harris, and J. Grimes, "Improving ssl warnings: Comprehension and adherence," in *Proceedings of the Conference on Human Factors and Computing Systems*, 2015.
[35] R. Sleevi, "Out of date Chrome results in ERR_CERTIFICATE_TRANSPARENCY_REQUIRED for Symantec operated sites," November 2016, https://bugs.chromium.org/p/chromium/issues/detail?id=664177.
[36] "Your connection is not private - NET::ERR_CERTIFICATE_TRANSPARENCY_REQUIRED," November 2016, https://productforums.google.com/forum/#!topic/chrome/YLX1__NgPj0.
[37] "your connection is not private," January 2017, https://productforums.google.com/forum/#!topic/chrome/nSt-w4eaefM.
[38] "Certificate Transparency Log Policy," October 2016, https://www.chromium.org/Home/chromium-security/certificate-transparency/log-policy.
[39] R. Sleevi, "Upcoming Log Removal: PuChuangSiDa," May 2017, https://groups.google.com/a/chromium.org/d/msg/ct-policy/M-rhwDQ1h9E/uX3NL5ndAwAJ.
[40] "Upcoming Log Removal: Venafi CT Log Server," March 2017, https://groups.google.com/a/chromium.org/d/msg/ct-policy/KMAcNT3asTQ/UgJ70hvBBQAJ.
[41] "Venafi has produced inconsistent STHs," February 2017, https://groups.google.com/a/chromium.org/d/msg/ct-policy/ohtZ64gLN3I/q4nSPkrWCQAJ.
[42] "Certificate Transparency - Cloudflare 'nimbus2018' Log Server Inclusion Request," November 2017, https://bugs.chromium.org/p/chromium/issues/detail?id=780654#c14.
[43] J. Amann, O. Gasser, Q. Scheitle, L. Brent, G. Carle, and R. Holz, "Mission accomplished?: HTTPS security after diginotar," in *Proceedings of the 2017 Internet Measurement Conference*. ACM, 2017, pp. 325–340.
[44] "Configuring Nginx to send certificate transparency SCT," September 2017, https://community.letsencrypt.org/t/configuring-nginx-to-send-certificate-transparency-sct/41474.
[45] "Apache Module mod_ssl_ct," https://httpd.apache.org/docs/trunk/mod/mod_ssl_ct.html.
[46] C. Evans, C. Palmer, and R. Sleevi, "Public key pinning extension for HTTP," April 2015, https://tools.ietf.org/html/rfc7469.
[47] M. Biilmann, "Be afraid of HTTP Public Key Pinning (HPKP)," October 2016, https://www.smashingmagazine.com/be-afraid-of-public-key-pinning/.
[48] V. Lynch, "Industry experts agree: Don't use key pinning (HPKP)," August 2017, https://www.thesslstore.com/blog/industry-experts-say-dont-use-key-pinning-hpkp/.
[49] "Remove pinning for cryptocat," January 2015, https://bugs.chromium.org/p/chromium/issues/detail?id=446240.
[50] A. P. Felt, R. Barnes, A. King, C. Palmer, C. Bentzel, and P. Tabriz, "Measuring HTTPS adoption on the web," 2017.
[51] C. Palmer, "Intent to deprecate and remove: Public key pinning," October 2017, https://groups.google.com/a/chromium.org/d/msg/blink-dev/he9tr7p3rZ8/eNMwKPmUBAAJ.
[52] R. Sleevi, "Upcoming CT Log Shutdown: Aviator," November 2016, https://groups.google.com/a/chromium.org/d/msg/ct-policy/u87C79AY-E8/VM4K1v8qCgAJ.
[53] D. O'Brien, "Upcoming CT Log Removal: WoSign," January 2018, https://groups.google.com/a/chromium.org/d/msg/ct-policy/UcCqlxuz_1c/Mf_939xYAQAJ.
[54] R. Barnes, "[Trans] WGLC comments on draft-ietf-trans-6962-bis-24," January 2017, https://www.ietf.org/mail-archive/web/trans/current/msg02623.html.
[55] B. Laurie, P. Phaneuf, and A. Eijdenberg, "Certificate transparency over DNS," March 2016, https://github.com/google/certificate-transparency-rfcs/blob/master/dns/draft-ct-over-dns.md.
[56] A. Ayer, "STH pollination implementations," March 2017, https://www.ietf.org/mail-archive/web/trans/current/msg02807.html.
[57] G. Edgecombe, "Wosign log failure to incorporate entry within the MMD," December 2017, https://groups.google.com/a/chromium.org/d/msg/ct-policy/-eV4Xe8toVk/pC5gSjJKCwAJ.
[58] D. O'Brien, "Upcoming CT Log Removal: WoSign," January 2018, https://groups.google.com/a/chromium.org/d/msg/ct-policy/UcCqlxuz_1c/Mf_939xYAQAJ.
[59] E. Messeri, "Privacy implications of Certificate Transparency's DNS-based protocol," November 2017, https://docs.google.com/document/d/1DY2OsrSJDzlRHY68EX1OwQ3sBIbvMrapQxvANrOE8zM/edit#.
[60] M. Kliemani, "Privacy, redaction and certificate transparency," May 2016, https://www.symantec.com/connect/blogs/privacy-redaction-and-certificate-transparency.
[61] A. P. Felt, E. Ha, S. Egelman, A. Haney, E. Chin, and D. Wagner, "Android permissions: User attention, comprehension, and behavior," in *Proceedings of the Eighth Symposium on Usable Privacy and Security*, ser. SOUPS '12. New York, NY, USA: ACM, 2012, pp. 3:1–3:14. [Online]. Available: http://doi.acm.org/10.1145/2335356.2335360
[62] S. Motiee, K. Hawkey, and K. Beznosov, "Do Windows users follow the principle of least privilege?: Investigating user account control practices," in *Proceedings of the Sixth Symposium on Usable Privacy and Security*, ser. SOUPS '10. New York, NY, USA: ACM, 2010, pp. 1:1–1:13. [Online]. Available: http://doi.acm.org/10.1145/1837110.1837112
[63] S. Egelman, L. F. Cranor, and J. Hong, "You've been warned: An empirical study of the effectiveness of web browser phishing warnings," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '08. New York, NY, USA: ACM, 2008, pp. 1065–1074. [Online]. Available: http://doi.acm.org/10.1145/1357054.1357219
[64] C. Bravo-Lillo, S. Komanduri, L. F. Cranor, R. W. Reeder, M. Sleeper, J. Downs, and S. Schechter, "Your attention please: Designing security-decision uis to make genuine risks harder to ignore," in *Proceedings of*

the Ninth Symposium on Usable Privacy and Security, ser. SOUPS '13. New York, NY, USA: ACM, 2013, pp. 6:1–6:12. [Online]. Available: http://doi.acm.org/10.1145/2501604.2501610

[65] C. Bravo-Lillo, L. Cranor, S. Komanduri, S. Schechter, and M. Sleeper, "Harder to ignore? revisiting pop-up fatigue and approaches to prevent it," in *10th Symposium On Usable Privacy and Security (SOUPS 2014)*. Menlo Park, CA: USENIX Association, 2014, pp. 105–111. [Online]. Available: https://www.usenix.org/conference/soups2014/proceedings/presentation/bravo-lillo

[66] D. Akhawe, B. Amann, M. Vallentin, and R. Sommer, "Here's my cert, so trust me, maybe?: Understanding TLS errors on the web," in *Proceedings of the 22Nd International Conference on World Wide Web*, ser. WWW '13. New York, NY, USA: ACM, 2013, pp. 59–70. [Online]. Available: http://doi.acm.org/10.1145/2488388.2488395

[67] S. Fahl, Y. Acar, H. Perl, and M. Smith, "Why Eve and Mallory (also) love webmasters: A study on the root causes of SSL misconfigurations," in *Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security*, ser. ASIA CCS '14. New York, NY, USA: ACM, 2014, pp. 507–512. [Online]. Available: http://doi.acm.org/10.1145/2590296.2590341

[68] B. VanderSloot, J. Amann, M. Bernhard, Z. Durumeric, M. Bailey, and J. A. Halderman, "Towards a complete view of the certificate ecosystem," in *Proceedings of the 2016 ACM on Internet Measurement Conference, IMC 2016, Santa Monica, CA, USA, November 14-16, 2016*, 2016, pp. 543–549. [Online]. Available: http://dl.acm.org/citation.cfm?id=2987462

[69] J. Gustafsson, G. Overier, M. F. Arlitt, and N. Carlsson, "A first look at the CT landscape: Certificate transparency logs in practice," in *PAM*, 2017.

[70] Z. Durumeric, J. Kasten, M. Bailey, and J. A. Halderman, "Analysis of the HTTPS certificate ecosystem," in *Proceedings of the 2013 Conference on Internet Measurement Conference*, ser. IMC '13. New York, NY, USA: ACM, 2013, pp. 291–304. [Online]. Available: http://doi.acm.org/10.1145/2504730.2504755

[71] O. Levillain, "A study of the TLS ecosystem," Theses, Institut National des Télécommunications, Sep. 2016. [Online]. Available: https://tel.archives-ouvertes.fr/tel-01454976

[72] Z. Durumeric, E. Wustrow, and J. A. Halderman, "Zmap: Fast internet-wide scanning and its security applications," in *Presented as part of the 22nd USENIX Security Symposium (USENIX Security 13)*. Washington, D.C.: USENIX, 2013, pp. 605–620. [Online]. Available: https://www.usenix.org/conference/usenixsecurity13/technical-sessions/paper/durumeric

[73] D. Naylor, A. Finamore, I. Leontiadis, Y. Grunenberger, M. Mellia, M. Munafò, K. Papagiannaki, and P. Steenkiste, "The cost of the "s" in HTTPS," in *Proceedings of the 10th ACM International on Conference on Emerging Networking Experiments and Technologies*, ser. CoNEXT '14. New York, NY, USA: ACM, 2014, pp. 133–140. [Online]. Available: http://doi.acm.org/10.1145/2674005.2674991

[74] L. Chuat, P. Szalachowski, A. Perrig, B. Laurie, and E. Messeri, "Efficient gossip protocols for verifying the consistency of certificate logs," in *2015 IEEE Conference on Communications and Network Security (CNS)*, Sept 2015, pp. 415–423.

[75] S. Eskandarian, E. Messeri, J. Bonneau, and D. Boneh, "Certificate transparency with privacy," *Proceedings on Privacy Enhancing Technologies*, vol. 2017, no. 4, pp. 329–344, 2017.