Fixing HTTPS Misconfigurations at Scale: An Experiment with Security Notifications

Eric Zeng
Google, University of Washington
ericzeng@cs.washington.edu

Frank Li
University of California, Berkeley
frankli@cs.berkeley.edu

Emily Stark
Google
estark@chromium.org

Adrienne Porter Felt
Google
felt@chromium.org

Parisa Tabriz
Google
parisa@chromium.org

Abstract

HTTPS is vital to protecting the security and privacy of users on the Internet. As the cryptographic algorithms and standards underlying HTTPS evolve to meet emerging threats, website owners are responsible for updating and maintaining their HTTPS configurations. In practice, millions of hosts have misconfigured and insecure configurations. In addition to presenting security and privacy risks, misconfigurations can harm user experience on the web, when browsers show warnings for deprecated and outdated protocols.

We investigate whether sending direct notifications to the owners of misconfigured sites can motivate them to fix or improve HTTPS misconfigurations, such as outdated ciphersuites or certificates that will expire soon. We conducted a multivariate randomized controlled experiment testing multiple variations of message content through two different notification channels. We find that security notifications alone have a moderate impact on remediation outcomes, similar to or less than notifications for other types of security vulnerabilities. We discuss how notifications can be used in conjunction with other incentives and outreach campaigns, and identify future directions for improving the security of the HTTPS ecosystem.

1 Introduction

HTTPS is fundamental to the security of the web, ensuring the confidentiality and integrity of traffic between clients and web servers. Its adoption has risen steadily over the past several years, with more than half of top websites supporting the protocol by 2017 [14]. However, HTTPS adoption alone is not enough to provide users with strong security and privacy guarantees. Site owners must continuously maintain their HTTPS deployments, by updating configurations and obtaining new certificates as cryptographic and protocol standards evolve to address new security concerns. Outdated and insecure sites may be penalized by browsers and search engines, leading to poor user experiences and warning fatigue [2].

Despite these negative consequences, HTTPS misconfigurations occur frequently in the wild [2, 3]. For example, millions of websites serve incomplete certificate chains or deploy outdated and insecure versions of the TLS protocol (Section 5.1). Members of the HTTPS ecosystem react in different ways that may encourage site owners to fix misconfigurations:

1. **Browser warnings.** Web browsers can protect users from security misconfigurations by removing support for insecure configurations and using warnings to block users from accessing misconfigured sites. Site owners may quickly learn when their site is causing a browser warning and fix the misconfiguration.
2. **Broad outreach.** Web browsers, researchers, and certificate authorities can conduct broad public outreach about misconfigurations using blog posts, newsletters, or other communications.
3. **Security notifications.** Researchers can directly send messages to the administrators of misconfigured websites, using contact points found in public databases or opt-in notification channels.

Browser warnings and outreach are commonly used in practice [30], but these approaches have drawbacks. Warnings can inconvenience users and cause confusion or warning fatigue, making future warnings less effective. Outreach alone may not reach all affected site owners.

Security notifications are an attractive alternative because they could in theory directly target owners of misconfigured sites without affecting end users. However, it is an open question whether notifications are effective at increasing the remediation of HTTPS misconfigurations. While notifications have been used with some success for vulnerable and hijacked systems [6, 7, 12, 23–25, 34–36], the only example of using direct notifications for an HTTPS security issue was conducted by Durumeric et al. [12], who notified servers vulnerable to the Heartbleed bug, observing significant increases in patching when notified. It remains unstudied whether this approach is similarly effective for less severe and infamous types of HTTPS misconfigurations.

In this work, we investigate whether security notifications are an effective way to encourage website owners to remediate HTTPS misconfigurations. We evaluate the effectiveness of security notifications in two different contexts: (1) a standalone notification campaign for common misconfiguration, such as outdated ciphersuites and incomplete certificate chains, and (2) a notification campaign about distrusted Symantec certificates, in conjunction with public outreach and planned browser UI changes.

In our standalone notification campaign, we further explore best practices for sending these notifications via a randomized controlled multivariate experiment for several classes of HTTPS misconfigurations. In our experiment, we test two different notification channels: (1) emails to WHOIS contacts, and (2) emails sent via Google Search Console[1], a free opt-in service for receiving Google's diagnostics on websites. We also experiment with different message constructions, varying the message's language, persuasive approach, and subject line. Across different treatment groups and controls, we monitor the remediation rates to identify the most effective notification conditions.

Contributions

- We conduct the first controlled experiments testing the effectiveness of direct notifications for HTTPS misconfigurations. We find that direct notifications have a limited but statistically significant effect on remediation.
- We compare two channels for these security notifications, Google Search Console and WHOIS contact emails, and find evidence that Search Console is more effective.
- We compare variations on message construction, and detect no significant impact on remediation rates as a result of our variations.
- We conduct a survey of website owners who received our notifications, and we characterize their reactions to and understanding of our messages.
- Based on our findings, we recommend methods for improving the health of the HTTPS ecosystem.

2 Background

HTTPS protects the confidentiality and integrity of web traffic from network attackers. Websites using HTTPS have a public key, which can be used to encrypt communications between clients and the website's servers. To establish trust in the site's public key, the site must be issued a certificate by a trusted certificate authority (CA). In order to establish a successful connection, clients verify that a valid chain of trust exists between the site's certificate and the root certificate for the issuing certificate authority. Clients also perform a number of additional checks, such as checking that the server's certificate is not expired and that the requested hostname matches the name in the certificate. When a web browser cannot successfully validate a server's certificate chain, it will show the user a full-screen error page.

When configured correctly, HTTPS prevents network attackers from eavesdropping on or modifying connections. However, in the real world, many server-side configurations are out-of-date or invalid. Server-side misconfigurations can prevent users from accessing the site or cause clients to establish a less secure connection that is more vulnerable to attack. We identify three classes of misconfigurations, and we consider examples of each of them in our experiments.

Outdated TLS Configurations. TLS and its predecessor SSL are the cryptographic protocols underlying HTTPS [10]. When web servers do not support the latest version of TLS or modern cryptographic settings, clients connect over outdated connections that leave users vulnerable to attack [26, 33].

- **Outdated TLS Version:** TLS 1.2 was the latest version of TLS at the time of our experiment. Sites that do not support TLS 1.2 are less secure because of known weaknesses in older versions, and because they don't support the latest ciphersuites.
- **Outdated TLS Ciphersuite:** Currently, it is recommended that servers prefer ciphersuites with Authenticated Encryption with Associated Data (AEAD) [27]. Sites may be configured to use weaker ciphersuites for encrypting, signing, and authenticating connections, some of which are known to be vulnerable to attacks.

Certificate Misconfigurations. If the website provides an invalid or malformed certificate that web browsers cannot validate, users will see full-page browser warnings.

- **Incomplete Certificate Chain:** Websites serve a set of certificates that the client uses to build a chain to a trusted root certificate. When a website does not serve all the necessary certificates to build a chain, some web browsers will use cached or dynamically fetched intermediate certificates to complete the chain [8], but browsers that do not will fail to validate the certificate. This issue primarily affects users of Mozilla Firefox and older versions of Google Chrome for Android.

Soon-to-Be Invalid Certificates. Websites must update their certificates periodically to account for expiration and the changing requirements of web browsers. Ideally, website owners should replace certificates before they become invalid. However, if site owners are unaware that their certificates will become invalid, browsers and researchers could remind site owners to renew their certificates before users are impacted.

- **Certificate Expiring Soon:** All certificates will eventually expire and become invalid. If a website owner forgets to renew a certificate before its expiration date, users will see full-page browser warnings.

[1] `https://www.google.com/webmasters/tools/home`

- **Certificate Distrusted Soon:** Browsers sometimes distrust certificates due to certificate misissuances or security breaches at CAs or hosting providers, which may have compromised private keys for signing certificates or establishing connections. Site owners must obtain new certificates before the distrust date to avoid errors.

Site owners can fix many HTTPS misconfigurations by obtaining a new certificate, adjusting a server-side configuration file, or upgrading server software. However, HTTPS misconfigurations are widespread (Section 5.1), suggesting that current efforts for informing site owners of vulnerabilities and deprecations are insufficient, or that site owners feel little incentive to do so. Previous work by Fahl et al. [13] and Krombholz et al. [22] indicates that many site owners don't believe HTTPS to be important, don't realize that misconfigurations can cause warnings, and have difficulty selecting secure ciphersuites and certificate configurations.

Web browsers have historically phased out insecure HTTPS configurations by making changes to browser UI, such as browser warnings, and then informing site owners of the change via blog posts and other communications [30]. Browser warnings protect end users by alerting them to real attacks, but they can also result in false positives for sites that are just misconfigured, causing confusion and warning fatigue. To minimize the impact on end users, we investigate whether direct notifications can be used to encourage site owners to fix HTTPS misconfigurations.

3 Related Work

3.1 Measuring HTTPS Misconfigurations

HTTPS misconfigurations are a well-known phenomenon in the security and measurement communities. By conducting large-scale scans of HTTPS servers across the Internet, researchers have created a comprehensive picture of HTTPS certificates and configurations, including a substantial number of misconfigurations.

In one of the first large-scale measurements of real-world TLS usage, Holz et al. [20] uncovered many misconfigured sites in the Alexa Top Million with weak ciphersuites and expired, self-signed, and invalid certificates.

Subsequent research has improved the completeness of the scans, and included deeper analysis of misconfigurations and vulnerabilities. Heninger et al. [18] scanned the entire IPv4 Internet for HTTPS servers, and discovered hundreds of thousands of hosts reusing keys and using low entropy keys. Durumeric et al. [11] scanned the HTTPS certificate ecosystem across the IPv4 address space, and discovered that 55K certificates used factorable RSA keys, 33K certificates were signed using MD5 for hashing, and 12.8% of all certificates were invalid or had an invalid, incomplete, or misordered certificate chain.

However, research based on IPv4 scans often captured misconfigured web servers that most web users never access, such as embedded devices that serve self-signed certificates. Rather than measuring every reachable host on the Internet, Acer et al. [2] used HTTPS error reports from the Google Chrome browser to identify misconfigurations that had the greatest impact on users. One example that helped motivate our study was that 35.8% of errors seen by Android users were caused by incomplete certificate chains.

3.2 Security Notifications

In the last six years, the security community has explored using email notification campaigns to inform parties affected by security issues. Several studies have found that notifying webmasters of compromised and hijacked websites substantially increases the remediation of infections [7, 25, 36], with Li et al. reporting a 50% increase in the likelihood of remediation [25], and Vasek et al. reporting that 62% of notified sites remediated compared to 45% of control sites [36]. Other studies reported increased remediation rates after notifying servers with vulnerabilities such as DDoS amplifiers [23, 24], XSS [34, 35], Heartbleed [12], and firewall misconfigurations [24].

These studies have revealed numerous challenges in reaching the administrators of vulnerable hosts, observing email bounce rates for WHOIS contacts at around 6-9%, message read rates at around 5-15%, and low engagement with feedback mechanisms [34, 35]. Stock et al. found that invalid points of contact, spam filtering, and initial mistrust in the message likely accounted for a significant fraction of failed communications and low engagement [34].

These studies also helped surface potential best practices for crafting and sending notifications. Li et al. [24], Cetin et al. [7], and Vasek et al. [36] all found that more comprehensive messages resulted in higher patch rates. However, experiments with techniques like sending repeat notifications [24, 35] using senders with high reputation [7, 34], and varying email formats [34] produced conflicting or inconclusive results.

Little work has examined using security notifications in the domain of HTTPS misconfigurations. In 2014, Durumeric et al. ran a notification campaign for the Heartbleed vulnerability. This vulnerability had broad awareness, with stories appearing in the news [15]. However, browsers did not incentivize patching by using warnings to block vulnerable sites. Their notification campaign increased patch rates by 47%. It is unclear though if their results generalize beyond Heartbleed to other types of HTTPS misconfigurations.

4 Methodology: Notifications Experiment

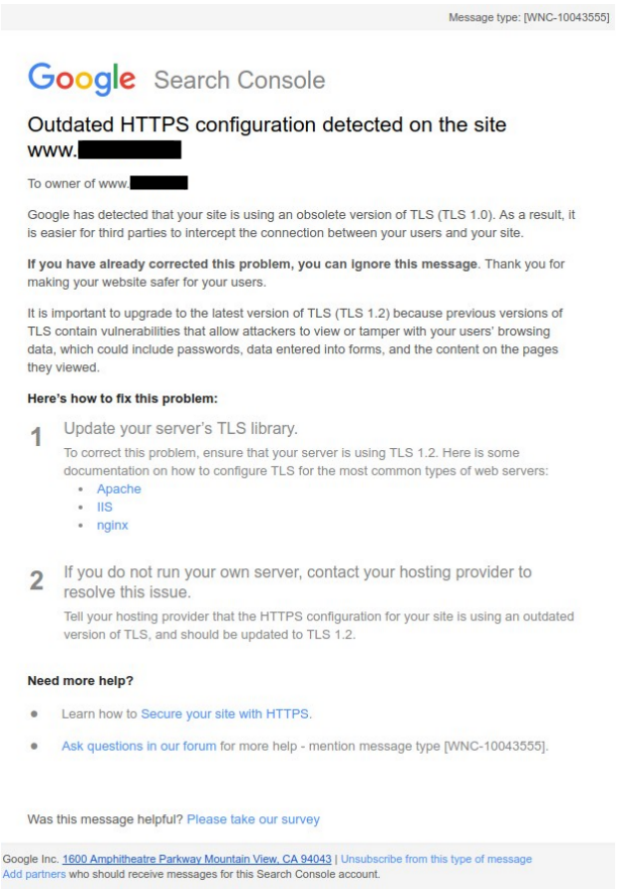
We conducted a randomized controlled experiment to investigate whether security notifications can encourage site own-

ers to remediate HTTPS misconfigurations, and factors that might affect the effectiveness of the notifications. We sought to answer the following research questions:

- Are notifications an effective tool for encouraging site owners to fix HTTPS misconfigurations?
- What effect do notifications have in the absence of other outreach or browser warnings?
- How does notification effectiveness differ across misconfiguration types?
- Do the sender and the channel of the notification influence the effectiveness of the notification?
- Does the message content (language, subject line, framing) influence the effectiveness of the notification?
- How do site owners perceive and react to notifications about HTTPS misconfigurations?

In our experiment, we selected types of HTTPS misconfigurations that were common on the Internet, but not yet targeted by browser warnings or broad outreach campaigns. We took a sample of websites with these HTTPS misconfigurations, and assigned them to different groups; either the control group which did not receive notifications, or a treatment group which received one variation of the message.

Our messages were sent via email, and explained who we (the senders) were, what HTTPS misconfiguration was detected, why the misconfiguration was an issue, and how specifically the site owner could resolve the problem. An example of a notification message for the outdated TLS version misconfiguration, sent via Google Search Console, is shown in Figure 1. We provide additional examples of the incomplete chain message sent via WHOIS, in Appendix B.


Message type: [WNC-10043555]

Google Search Console

Outdated HTTPS configuration detected on the site www.[redacted]

To owner of www.[redacted]

Google has detected that your site is using an obsolete version of TLS (TLS 1.0). As a result, it is easier for third parties to intercept the connection between your users and your site.

If you have already corrected this problem, you can ignore this message. Thank you for making your website safer for your users.

It is important to upgrade to the latest version of TLS (TLS 1.2) because previous versions of TLS contain vulnerabilities that allow attackers to view or tamper with your users' browsing data, which could include passwords, data entered into forms, and the content on the pages they viewed.

Here's how to fix this problem:

1 Update your server's TLS library.

To correct this problem, ensure that your server is using TLS 1.2. Here is some documentation on how to configure TLS for the most common types of web servers:

- Apache
- IIS
- nginx

2 If you do not run your own server, contact your hosting provider to resolve this issue.

Tell your hosting provider that the HTTPS configuration for your site is using an outdated version of TLS, and should be updated to TLS 1.2.

Need more help?

- Learn how to Secure your site with HTTPS.
- Ask questions in our forum for more help - mention message type [WNC-10043555].

Was this message helpful? Please take our survey

Google Inc. 1600 Amphitheatre Parkway Mountain View, CA 94043 | Unsubscribe from this type of message
Add partners who should receive messages for this Search Console account.

Figure 1: Example notification message for the outdated TLS version misconfiguration (site name redacted).

4.1 Misconfiguration Detection

To identify a set of misconfigured websites to notify, we built a system that detects HTTPS misconfigurations. Our detector analyzes data from Googlebot, Google's web crawler [17], which stores the HTTPS certificates and connection parameters for each HTTPS site that it crawls. Our detector identifies sites with HTTPS misconfigurations (described in Section 2) using the following rules:

1. *Incomplete Certificate Chain*: If a website served a single certificate which was not directly signed by a trusted root certificate in the Mozilla root store [1], we consider the site's certificate chain to be incomplete[2].
2. *Certificate Expiring Soon*: We consider a site's certificate to be expiring soon if the date in its *Not After* field [21] is within the next two weeks at the time of the scan. We omit certificates that will expire within one week to account for a delay of several days between the scan and when we send notifications, to avoid sending notifications to certificates that already expired.
3. *Outdated TLS Version*: If Googlebot established a connection to a site using SSL 3.0, TLS 1.0, or TLS 1.1, we consider the site's TLS version to be outdated.
4. *Outdated Ciphersuite*: If the Googlebot crawler established a TLS connection to a site using a non-AEAD ciphersuite, we consider its ciphersuite to be outdated. (See Appendix A for a list of AEAD ciphersuites.)

After a site is notified about a misconfiguration, our detector actively scans the site's status on a daily basis, allowing us to observe changes in the state of its HTTPS deployment.

4.2 Experiment Design: Variables

To explore different factors that could influence the effectiveness of notifications, we varied the notification messages along the following dimensions.

4.2.1 Notification Channel and Sender

We notified the owners of misconfigured sites using two different notification channels:

Google Search Console. Search Console is a free opt-in service for website owners, who can receive notices from

[2]This detection logic does not capture all the ways in which a server's certificate chain can be incomplete, but we use this method because it is simplest and does not suffer from false positives.

Google about search indexing tips, website performance, and security issues. For websites that we contacted via Search Console, emails were sent to the addresses registered with the Search Console accounts. These emails were sent following Google's standard messaging practices: they were styled using Google's standard email formatting with official branding, and were sent from a *google.com* email domain.

WHOIS emails. WHOIS is a public database containing ownership information and points of contact for domains, IP addresses, and ASNs[3]. This channel is available to the security community at large, and has been used in prior notification studies [12, 24, 34, 35]. We obtained WHOIS email contacts on websites using RiskIQ's PassiveTotal API[4] [31]. We sent email messages to WHOIS contacts from a UC Berkeley domain, with *security-notifications* as the sender. Our messages contained the same content as the Search Console messages and were similarly styled, but with the branding of our authors' universities and a link to a supplementary website (also hosted on a UC Berkeley domain) with further details about the misconfigurations and our study.

A limitation we faced was that we were unable to separate the notification sender and channel, because Google did not permit us to send messages with UC Berkeley branding via Search Console, nor were we permitted to send the Google-branded emails via WHOIS. While this makes it harder to disentangle whether an effect comes from the channel or the sender, it also reflects the practical constraints for those hoping to send security notifications.

4.2.2 Message Language

Li et al. [25] found that vulnerability notification messages in English were more effective than messages translated into the native language of the contact. Follow-up surveys with system administrators indicated that many expected English messages (particularly when arriving from the authors' US universities), and initially thought the translated messages were spam. This finding was ultimately limited though by the diversity of languages explored (four European languages) and the experiment's population size.

We attempted to replicate this finding in our experiment. We subdivided our Search Console notifications into two groups: one that received English messages, and the other that received translated ones. Messages were translated into 39 languages by an internationalization team at Google. We were unable to obtain copies of the translations for our WHOIS notifications.

[3]We accessed WHOIS prior to the implementation date of the European Union's General Data Protection Regulation (GDPR) [28]. Thus, we do not believe GDPR had a significant impact on our WHOIS experiment, although it could impact future similar experiments.

[4]We used the default contact provided by the PassiveTotal API, which prioritizes technical and admin contacts over registrants and other contacts.

4.2.3 Message Framing

How should we frame the message to best grab the attention of site owners, motivate them to take corrective steps, and provide sufficient resources to solve the issue? To explore this question, we considered two variations of the notification message text:

User focus (Variant A). This variant tests the hypothesis that the notification is more persuasive when it emphasizes the misconfiguration's impact on users - the threat of data tampering, browser warnings impeding access to the site, harm to site reputation, etc. While we state what the misconfiguration is, we do not extensively explain its technical details.

Technical focus (Variant B). Another hypothesis is that messages should include more technical detail about the misconfiguration to help site owners understand why it poses a security risk. While we mention the immediate user-facing effects (like users seeing a browser warning), we don't discuss the ways HTTPS protects users in general.

For our experiments, we explore these variations on the message for the incomplete certificate chain misconfiguration, where the misconfiguration technical details were complicated enough to warrant the variations. For the other misconfigurations, the issues were more technically straightforward, making it challenging to create meaningfully different versions. We reproduce the message text in Appendix B.

4.2.4 Subject Line

Similar to the message's framing, the nature of the email subject may affect how much attention it attracts from recipients and how clearly its purpose is conveyed.

General (Variant A). We hypothesized that a more general subject would make it clearer that it is a security/HTTPS issue, increasing the chance that the message is forwarded to the correct person. The subject line for an outdated ciphersuite notification was: "Outdated HTTPS configuration detected on the site $\{url\}$"

Specific (Variant B). We hypothesized that more specific subject lines could help the technical site owners quickly identify and fix the problem. The subject line for an outdated ciphersuite notification was: "Outdated TLS ciphersuites are being used on the site $\{url\}$"

We tested two variants of email subjects for the outdated TLS version and outdated ciphersuite misconfigurations. We did not test this variable for sites with expiring certificates, as there isn't a more specific or general terminology, or for sites with incomplete certificate chains, to avoid excessively subdividing the population and lowering the statistical power of our tests.

4.3 Experiment Design: Treatment Groups

For the population of sites affected by each misconfiguration, we randomly selected groups of 1000 sites to receive each

Misconfig Type	Sender	Language	Framing	Subject
Incomplete Chain	✓	SC only	✓	✗
Outdated Ciphersuite	✓	SC only	✗	✓
Outdated TLS Version	✓	SC only	✗	✓
Cert Expires Soon	✓	SC only	✗	✗

Table 1: Random group assignment constraints. A check mark indicates that both levels of the variable were tested for that misconfiguration type.

different treatment, with one group designated as the control group, receiving no notifications. Our treatments were the factorial combinations of our experiment variables, under the constraints described in Table 1.

The notification channel could not be entirely randomly assigned, because not all sites were registered with Search Console and not all sites had WHOIS contacts. Instead, we randomly sampled the experiment group from the population of sites addressable through that channel. Since the channels a site can be reached by could affect how likely that site is to remediate, when analyzing the notification channel variable we ensure that all sites in the comparison are reachable by the same channel(s). (See Sections 6.1.1-6.1.3)

When selecting experiment groups, we deduplicated websites that shared the same leaf certificate, choosing the misconfigured website that Googlebot crawled most recently, because a shared certificate may indicate that two different websites share the same operator.

We sent out notifications in two batches, with an initial pilot batch followed by a complete second batch. In the first batch, sent on December 11, 2017, we only notified sites with outdated TLS versions and ciphersuites, and expiring certificates. In total, we sent 5K WHOIS notifications and 10K Search Console notifications, and had 6K control sites. The second batch added sites with incomplete certificate chains, and was sent on March 15, 2018. We sent a total of 7K WHOIS notifications and 14K Search Console notifications, and had 8K control sites. We monitored notified sites for 56 days.

4.4 Experiment Design: Survival Analysis

To analyze the effect of the notifications on remediation rates, we used a family of statistical tools called survival analysis. Survival analysis enables comparisons on right-censored data: that is, data in which subjects are monitored over time for some "event", but the event may occur after we stop collecting data. This is typically used for medical studies where the "event" represents a subject's death, but in our case, the event refers to when a site remediates a misconfiguration. To account for right-censoring, statistics are computed on the estimated *survival function* or *hazard rate*: the probability of the event occurring as a function of time.

We can detect statistically significant differences in the remediation rates of two groups of sites using the *log-rank test*, a non-parametric hypothesis test that compares two survival functions [4]. We use this test to evaluate hypotheses about specific variables, such as "remediation rates are different for variation A versus variation B". We control for other variables by performing separate comparisons for each level of confounding variables, and ensuring that the comparisons are performed on sites drawn from similar populations (e.g. ensure all sites are registered with Search Console). When testing multiple hypotheses on the same population (e.g., sites in Search Console, sites with WHOIS contacts), the probability of false positives increases, which we correct for using the Holm-Bonferroni method [19].

However, the log-rank test does not produce an effect size, so to determine how much of an effect each variable has on remediation, we use the Cox Proportional Hazards model, a multiple regression model for survival analysis [5]. The Cox model approximates a baseline survival function, and then estimates the effect of each of the independent variables. The effect size for each variable is the hazard ratio: the ratio of the hazard rate for sites with the variable to the hazard rate of the baseline. For example, a hazard ratio of 1.4 can roughly be interpreted as showing that sites with that variable are 1.4 times more likely to remediate, compared to the baseline. We use the Cox model to determine the hazard ratio for variables that we found to be significant in the pairwise comparisons, as well as demographic factors that might affect the underlying baseline (see Section 5.2).

4.5 Qualitative Survey

To gauge the reactions of website owners to our notifications, we included a link to a short qualitative survey in all of our notification messages. We asked respondents whether they were aware of the misconfiguration prior to our notice, whether they had tried to fix it before, whether they found our messages trustworthy or useful, and if they would like to receive similar future notifications. The questions were a mix between yes/no questions, seven-point Likert scales, and short responses. The full list of questions is included in Appendix C.

A single coder analyzed short-response questions using an inductive method, generating descriptive codes (i.e., topic codes) [32] for each short response. We compared responses from WHOIS respondents to Search Console respondents by analyzing which codes did and did not exist in each group.

We compared Likert scale responses between WHOIS and Search Console respondents using the Mann-Whitney U test, a non-parametric test commonly used to check for differences between samples of Likert scale data [9]. We also calculate the common language effect size statistic [38], which is the probability that a random respondent in one group responded more positively than a random respondent in the other group.

We had a low response rate for the surveys. We received 25 responses from Search Console recipients, and 40 responses from WHOIS recipients, across both experiments. It is likely that there was response bias, due to many messages not reaching site owners (see Section 6.4). Therefore we limit our analysis to comparing types of responses between our WHOIS and Search Console groups, rather than making broad claims about notifications as a whole.

4.6 Ethical Considerations

To identify misconfigured websites, we used data extracted by Google's web crawler during its normal operations. This crawler visits websites at a limited rate, so it should not have caused undue traffic load on web servers. Additionally, as publicly documented [16], the crawler advertised a Google-specific user agent, and respected resource crawl policies listed in `robots.txt` files and robots HTML meta tags.

We followed guidelines and best practices developed by prior work [6, 7, 12, 24, 25, 34–36] for sending notifications. All notifications were sent to contacts that either explicitly opted into a service (e.g., Search Console) or were publicly listed (e.g., WHOIS). We respected requests to opt-out of our notifications and responded to all recipients who contacted us. Our email notifications were sent from mail servers with valid SPF and PTR DNS records to identify them as valid senders. Additionally, we provided a web page with additional information about our notification campaign, hosted on an HTTPS web server on a UC Berkeley domain.

Our survey was reviewed by the UC Berkeley IRB, and was determined to be exempt.

4.7 Limitations

Because we use the Googlebot crawler to find misconfigured sites, our dataset is not fully representative of the Internet, as not all sites have been indexed by Googlebot and the crawler does not visit every indexed site every day. However, we were still able to find millions of websites with HTTPS misconfigurations across the web (Section 5.1).

We could not control for other campaigns or events that could have influenced sites to fix their misconfigurations, but we are not aware of any other major outreach campaigns about this set of misconfigurations that ran at the same time as our campaign.

When forming experiment groups, we are unable to conclusively determine whether two sites are administered by the same entity using WHOIS and Googlebot data. It is possible that notifications for sites in one group could affect another. For example, if a single site owner has sites in a control group and an experiment group, the notification for the experiment group could cause them to remediate both sites.

Misconfiguration	# of Sites	Registered in Search Console	
Outdated TLS Version	1,276,696	55,381	(4.34%)
Outdated Ciphersuite	5,845,075	276,244	(4.73%)
Incomplete Chain	702,736	41,556	(5.91%)
Expires Soon	553,301	59,629	(10.78%)

Table 2: Number of HTTPS misconfigurations by type, in websites crawled by Googlebot, Nov. 2017

	Alexa Top 1M		Cisco Top 1M	
Misconfiguration	Dec '17	Mar '18	Dec '17	Mar '18
Outdated Ciphersuite	1.8%	1.5%	4.3%	4.0%
Outdated TLS Version	2.1%	2.1%	2.3%	2.3%
Cert Expires Soon	1.9%	2.1%	0.8%	0.9%
Incomplete Chain	—	2.3%	—	1.7%

Table 3: Percent of notified sites listed in the top 1M sites as determined by Alexa and Cisco Umbrella, split by the first and second rounds of the experiment.

5 Results: Measurements

First, we present our measurements of misconfigurations across the web, and the demographics of misconfigured sites, including their popularity, size, age, and hosting providers.

5.1 Misconfiguration Prevalence

Outdated TLS versions and ciphersuites were the most common misconfigurations in our scans. We detected far fewer sites misconfigured with certificate issues. Table 2 shows how many misconfigured sites we detected on sites crawled by Googlebot in the first week of November 2017.

5.2 Site Demographics

We conducted a demographic analysis to better understand the characteristics of the misconfigured sites.

Site popularity. Most sites with HTTPS misconfigurations are relatively small in traffic and user base. We looked up the number of misconfigured sites in our experiment that were listed in the June 21, 2018 snapshot of two datasets of popular sites: the Alexa Top Million[5] and Cisco Umbrella 1 Million[6] datasets. As shown in Table 3, we found that between 0.9% and 4.3% of sites were in the top 1 million, depending on the misconfiguration type and experiment round.

Site age and size. Across the misconfigurations we considered, site age and size are similar. Our Googlebot data included the number of URLs crawled on the domain, the date of the last significant update to the site, and the first time the

[5] `https://aws.amazon.com/alexa-top-sites/`

[6] `https://umbrella.cisco.com/blog/2016/12/14/cisco-umbrella-1-million/`

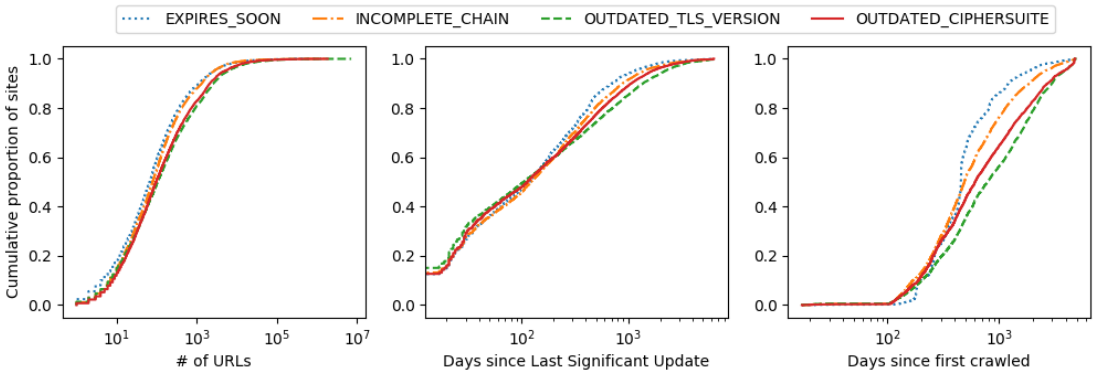


Figure 2: Demographic plots of the cumulative distribution of sites by site age and size, split by misconfiguration type, from our second experiment in March 2018. "# of URLs" is the number of URLs on the domain crawled by Googlebot in June 2018.

site was crawled. Figure 2 displays the cumulative distribution of sites for these metrics, split by misconfiguration.

Hosting Providers. Around half of the sites in our experiment sample were hosted by larger hosts: 14,805 of 29,015 domains in our second experiment were registered to organizations with 10 or more domains. We identified the hosting providers of sites by performing a WHOIS lookup on the IP addresses referred to by misconfigured hostnames.

We were interested in whether hosting providers would remediate misconfigurations across many of their domains as a result of our notifications, and our analysis showed that while some providers did fix most of their misconfigurations, many did not. For example, Cloudflare fixed all 30 incomplete chain misconfigurations, and Unified Layer fixed 47 of 59 incomplete chain misconfigurations. But for other providers like Amazon and GoDaddy, only a minority of domains fixed their misconfigurations. Table 4 shows remediation rates for the top 10 IP registrars by number of misconfigured sites.

6 Results: Notifications Experiment

Here, we present the results from our randomized controlled experiment. Unless otherwise specified, we present results from our second experiment iteration in March 2018, which we ran with more misconfiguration types.

6.1 Pairwise Log-Rank Test Comparisons

The results of our log-rank tests indicate that sites that receive notifications are more likely to remediate than those that don't, but we fail to detect significant differences in remediation as a result of variations on the message's content. We use a p-value threshold of $\alpha = 0.05$ when comparing two subsets of our study population, and the Holm-Bonferroni method for multiple comparison corrections (see Section 4.4). A full table of results is available in Appendix D.1, but we summarize each hypothesis tested below.

6.1.1 Sender: Google Search Console vs. Control

Search Console notifications can affect remediation. We compared remediation rates for sites that received Search Console notifications to control sites. We ensured that control sites were registered with Search Console to control for population factors. We detected a significant difference for outdated TLS versions and incomplete chains ($p \approx 0$ for both), but not for outdated ciphersuites and certificates expiring soon.

6.1.2 Sender: University + WHOIS vs. Control

For WHOIS notifications, we failed to detect significant changes in remediation behavior for all misconfigurations. We compared sites that received WHOIS notifications from the UC Berkeley alias to control sites. We ensured that control sites have a reachable WHOIS contact, to control for population factors. However, we note that we did detect that WHOIS notifications had an effect in our regression analysis (see Section 6.2).

6.1.3 Sender: Google Search Console vs. University+WHOIS

We failed to detect significant differences between Search Console and WHOIS notifications, suggesting that using a different channel does not have an effect on remediation rates. To determine this, we compared sites notified via Search Console to those notified via WHOIS. In both groups, we controlled for population by ensuring that all sites were registered in Search Console and had a WHOIS contact. Additionally, we controlled for language (English only), as we were not able to send translated messages over WHOIS.

We have two possible explanations for why we detected a difference between the Search Console group and the control group, but did not detect differences between the Search Console group and the WHOIS group, nor the WHOIS group and control group. First, the populations differ across com-

		Outdated TLS Version		Outdated Ciphersuite		Incomplete Chain	
Registrant	# sites	% of sites	% remediated	% of sites	% remediated	% of sites	% remediated
Amazon Technologies Inc.	958	16.70%	28.13%	19.83%	11.05%	48.12%	24.51%
GoDaddy.com, LLC	767	29.99%	9.57%	57.50%	2.95%	2.09%	37.50%
Hetzner Online GmbH	488	26.43%	14.73%	16.80%	8.54%	34.43%	30.95%
JPNIC-NET-JP	457	60.39%	9.42%	28.45%	1.54%	7.66%	22.86%
Unified Layer	441	4.54%	40.00%	3.85%	5.88%	13.38%	79.66%
Microsoft Corporation	372	8.87%	27.27%	62.37%	12.07%	20.16%	26.67%
Incapsula Inc	319	0%	0%	0.63%	50.00%	97.18%	49.35%
DigitalOcean, LLC	285	7.72%	27.27%	5.61%	12.50%	62.46%	24.72%
OVH	267	31.46%	17.86%	20.97%	3.57%	33.33%	25.84%
Amazon.com, Inc.	258	27.52%	15.49%	20.16%	11.54%	36.05%	31.18%

Table 4: Misconfigurations and remediation rates for the 10 largest hosting providers in our March 2018 experiment.

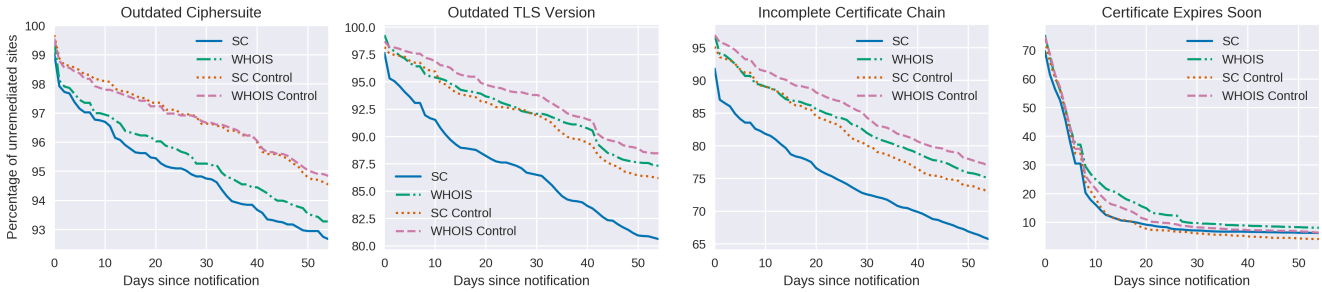


Figure 3: Comparison of remediation rates across notification channels, from March 2018.

parisons to control for confounding factors. Second, there could still be differences between the WHOIS group and both the Search Console group and control group that were too small to detect with our sample sizes.

6.1.4 Population: in Search Console vs. not in Search Console

For some misconfigurations, sites registered with Search Console are more responsive to notifications. Here, we held constant the WHOIS notification channel and computed separate comparisons for the A and B variations for message focus or subject line, in case of interaction effects. We found inconsistent results across misconfigurations, but some statistically significant differences in the expiring soon ($p \approx 0$) and incomplete chain (p=0.004 and p=0.002 in the A and B variations, respectively) misconfigurations. This suggests that Search Console registration is linked to a small increase in the likelihood of responding to notifications.

6.1.5 A/B Variations: Message Framing, Subject Line

We failed to detect any significant differences across variations in message framing. For incomplete chain misconfigurations, our A/B variation was on the message focus (user vs. technical), while for outdated TLS versions and outdated ciphersuites, we tested the subject line (general vs. specific). We did not A/B test the notification about certificates expiring soon. We compared separately for each notification channel and for translated and English-only messages, again in case of interaction effects.

6.1.6 Language: English-Only vs. Translated

In contrast with prior work [24], we did not observe differences in remediation between English-only messages and translation messages. We compared remediation rates for sites that received notifications translated to their preferred language with sites that received them in English regardless. We only compared Search Console messages, and we did separate comparisons for each A/B variation. We note that the language experiment conducted by Li et al. [24] was limited in language diversity and scale. Additionally, Li et al. attributed their observed difference to recipient surprise at receiving translated messages from US universities. This effect may not have applied for translated messages sent by Google, a company with international presence.

6.2 Survival Regression Analysis

The full table of results from the regression analysis is available in Appendix D.2. Here we report on statistically significant results (omitting those where we failed to detect a significant difference).

Sites that were notified were more likely to remediate, though the effect was not significant across all misconfiguration types. We saw that both WHOIS and Search Console messages had an effect on remediation rates. Com-

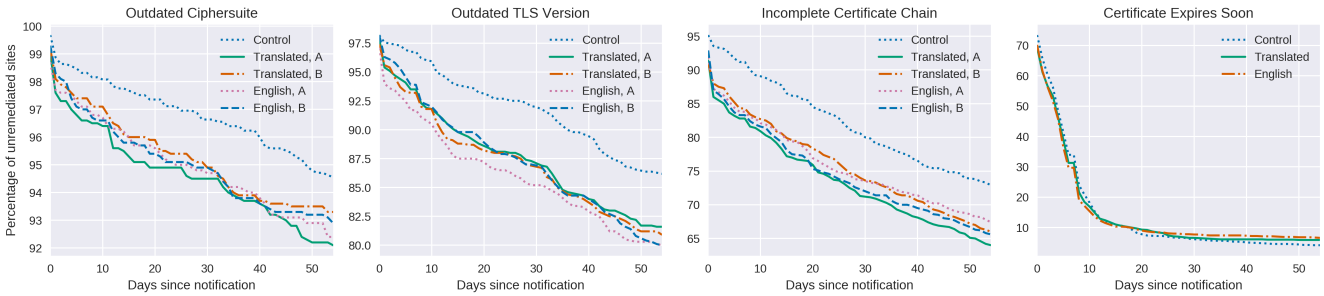


Figure 4: Comparison of remediation rates across message variations, for messages sent via Search Console, from March 2018.

pared to the control, sites notified via Search Console were 1.4, 1.48, and 1.35 times more likely to remediate, for the outdated ciphersuite, outdated TLS version, and incomplete chain misconfigurations (p=0.007, $p \approx 0$, $p \approx 0$), respectively. Messages sent via WHOIS were 1.56 and 1.25 times more likely to remediate for the outdated ciphersuite and incomplete chain misconfigurations (p=0.002, p=0.002).

Regardless of whether they received a notification, sites that were registered with Search Console were 1.38, 1.52, and 1.22 more likely to remediate for the outdated TLS version, incomplete chain, and expiring cert misconfigurations respectively (p=0.003, $p \approx 0$, $p \approx 0$). This provides more evidence to suggest that sites that registered for Search Console notifications are intrinsically more likely to take action and remediate, because they already opted into a service to improve their site.

Surprisingly, we found that sites that were in the Cisco Umbrella Top 1 Million were less likely to remediate than sites that were not. Sites in the Cisco Umbrella Top 1 million were 0.55, 0.64, and 0.73 times less likely to remediate (p=0.001, p=0.002, p=0.044) for the outdated ciphersuite, incomplete chain, and expired soon misconfigurations. We cannot say for certain why this is the case, but we speculate that it might be harder to fix misconfigurations in larger websites due to complexity or organizational inertia for security policies. For large organizations, it's also possible that we reached the wrong contact point, like a marketing or SEO specialist, who would need to route the misconfiguration report to someone in a system administrator role.

We did not find that the other demographic factors had a meaningful effect; while some of them were significant, the hazard ratios were very low (0.9999 or 1.0001), indicating that they had a negligible impact on remediation.

Note that while notifications increased the likelihood of remediation by 25-56%, due to the low baseline remediation rates, only 10% of sites in each group remediated that would not have otherwise.

Question	SC μ	WHOIS μ	CL	U	p
Trustworthiness	5.26	4.28	0.265	310.0	**0.020***
Acceptability	5.78	5.44	0.323	441.5	0.458
Future Messages	6.30	5.13	0.182	329.0	**0.027***

Table 5: Mann-Whitney U test comparing Search Console and WHOIS survey results for 7-point Likert scale questions.

6.3 Survey Analysis

First, we compared how people responded to the following questions, on a seven-point scale, between WHOIS and Search Console.

1. How trustworthy was our message?
2. How acceptable was it for us to detect the misconfiguration and notify you about the problem?
3. How interested would your organization be in receiving similar security and misconfiguration notifications in the future?

We hypothesized that there would be differences in trustworthiness, acceptability, and future messages, because site owners would perceive notifications and scans from Google differently than from a university, and because Search Console messages are solicited, while WHOIS emails are unsolicited. The Mann-Whitney U test in Table 5 shows that when comparing the response of a randomly selected Search Console respondent to a randomly selected WHOIS respondent, 27% of the time the Search Console respondent found the message more trustworthy than the WHOIS respondent (p=0.02), and 18% of the time they were more receptive to future messages (p=0.03).

Next, we analyzed short answer responses in the survey:

Trustworthiness. Most participants indicated that they trusted the notifications, citing the google.com or .edu domain in the email address.

> *"Clear, concise, e-mail domain matched domain behind links in the message and was a berkeley.edu domain."* – WHOIS41

A Search Console respondent also mentioned trusting the message because they remember opting into Search Console.

> *"I know the domain is added in search console and am familiar with incomplete cert chains so it made it easy."* – SC23

However, multiple WHOIS respondents distrusted the messages, and suspected that we were trying to phish or otherwise deceive them.

> *"Havent* [sic] *got a freaking clue who you are, so not convinced this isnt* [sic] *a very sophisticated phishing attempt"* – WHOIS11

Acceptability. Generally, respondents found it acceptable that we scanned for HTTPS misconfigurations, recognizing that it is all publicly accessible.

> *"literally anyone could and probably *has* already externally detected this issue. doing so and courteously informing the admin of the error is a rare kindness"* – WHOIS22

Disagreements. Despite generally positive feedback in terms of trustworthiness and acceptability, some respondents disagreed with our assessments and recommendations. Some respondents reported false positives (e.g., for certificates that were automatically renewed), and some disagreed with our recommendations, especially for ciphersuites.

> *"Removing 'obsolete' cipher suits* [sic] *can have dire consequences, and the list of cipher suits provided in the notification are not available on many platforms and is subjective."* – SC20

This highlights a general challenge in advancing HTTPS security, as some site owners will strongly prefer to continue supporting weaker ciphersuites and other features to preserve compatibility.

Other feedback. A couple of respondents requested tools to check whether their misconfigurations had been correctly remediated. Such tools exist for HTTPS configurations, such as the SSL Labs Server Test[7] and Mozilla Observatory[8], and could be included in future similar notifications.

6.4 Reachability Analysis

We were able to collect some limited data on whether our messages were received or read, which we present in Table 6.

Search Console. Around 60% of messages sent via Search Console were read by recipients. We experienced a couple of limitations using Search Console: first, only 85% of the sites we wanted to notify were sent notifications, due to per-user/site settings (for example, account settings in Search Console disallowing email messages). Second, as shown in Table 2, only 5-10% of misconfigured sites are registered with Search Console.

WHOIS. We did not implement trackers in our WHOIS emails, so we could only observe email bounces. We saw that about 3% of emails bounced, but cannot determine whether the remaining contacts saw our messages or not. In prior work, Stock et al. [34] observed that many messages were not read due to spam filters, inaccurate points of contact, and mistrust by recipients, in addition to bounces.

[7] `https://www.ssllabs.com/ssltest/`
[8] `https://observatory.mozilla.org/`

	Search Console			WHOIS		
Date	Sent	Read	Read %	Sent	Bounce	Bounce %
Dec 2017	8572	5394	62.93%	10000	323	3.2%
Mar 2018	11976	7265	60.66%	14000	393	2.8%

Table 6: Read and bounce rates for notification messages.

7 Case Study: Symantec Certificate Distrust Experiment

We had an opportunity to test our notification techniques in a separate, more urgent case of HTTPS misconfigurations, involving the distrust of certificates issued by the Symantec Certificate Authority. In 2017, the PKI community discovered serious concerns with Symantec's certificate issuance process, calling into question the trustworthiness of the certificates they issued. As a result, Mozilla [37] and Google [29] announced that they would be distrusting Symantec certificates in the Firefox and Chrome browsers, and would start showing full-screen browser warnings to users in April 2018.

We decided to run another randomized and controlled experiment, to investigate whether notifications have a beneficial effect in this context. In this case study, browsers were applying incentives for remediation by setting a deadline before warnings would be shown, and Google, Mozilla, and others were conducting significant public outreach.

7.1 Methodology

Using Googlebot data from March 7-10, 2018, we identified 715 sites with affected certificates that were registered with Search Console. We randomly split the sites into two groups: a group of 665 sites which we notified via Google Search Console on March 15, 2018, and a control group of 50 sites. Because of the time constraints imposed by the impending distrust event, we did not include other variables in the experiment, and we picked a smaller control group to maximize our outreach.

Message content. Because the fixed distrust timeline imposed tight time constraints, we did not experiment with message content but rather sent one message to all notified sites.

- The message was user-focused, focusing on the warnings that users would see and the need to replace certificates to avoid disruption to users.

- The message subject was "SSL/TLS certificate needs to be replaced for {*url*}".
- The message was sent to all affected sites in English.

7.2 Results

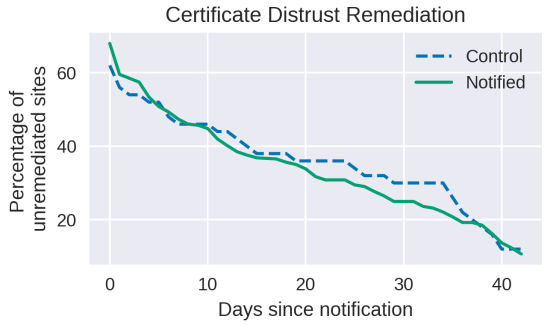


Figure 5: Remediation rates for sites in the Symantec experiment. Notifications were sent on March 15, 2018.

Using the log-rank test, we were unable to detect a significant difference in remediation between notified sites and control sites (p=0.80). However, overall remediation was good regardless. In the 5 days between our initial scan and when the notifications were sent, 30-40% of sites had already remediated, and by the end of the measurement period, only 12% of sites in both conditions still used affected certificates. We did not exclude sites from our analysis that had remediated between the time that we conducted our scan and the time that we sent the messages.

8 Discussion

Here we summarize our most salient results, distilling the lessons we have learned and suggesting directions for the security community moving forward.

8.1 Evaluating the Effectiveness of Security Notifications in the HTTPS Ecosystem

In this study, we experimented with using security notifications to drive the remediation of HTTPS misconfigurations.

In our main experiment, we ran a standalone notification campaign with no concurrent outreach campaign or browser changes, for a set of well-known misconfigurations, such as outdated ciphersuites, outdated versions of TLS, and incomplete certificate chains. We found that our notifications resulted in a statistically significant increase in remediation rates, with remediation increasing by 22%-52% for certain misconfigurations. However, the overall impact was limited; the absolute difference between control groups and experimental groups was less than 10%, and the total fraction of notified sites that remediated after 56 days ranged from 7% to 34%, depending on misconfiguration type.

In our case study, we ran a notification campaign about certificates from Symantec that were soon-to-be distrusted, in conjunction with a broad outreach effort by others in the HTTPS ecosystem, and impending browser warnings that would block users from visiting pages with distrusted Symantec certs. In contrast to the first experiment, we found that our notifications had no statistically significant effect on remediation, but 90% of all of the affected sites that we monitored had remediated after 40 days.

The different outcomes in these experiments suggest that the notifications can help drive remediation, but that the overall impact depends on the severity of the issue, the level of outreach and awareness, and the external incentives applied. In our main experiment, it appears that surfacing information to site owners did push some to fix their configurations, but that the lack of external incentives like browser warnings meant many site owners were not motivated to do so. In contrast, the outreach effort and warnings imposed by all major browsers in the certificate distrust case were so urgent and heavy-handed that it swamped the effect of our notifications.

8.1.1 Comparison with related studies

Our main experiment produced results similar to the Heartbleed experiment conducted by Durumeric et al. [12]. Like our experiment, there were no external incentives imposed by browsers to remediate Heartbleed, but unlike ours, notifications were sent in conjunction with broad awareness of the vulnerability. Their notifications improved remediation by a similar amount, with an 47% relative increase, and an absolute difference of 13% between their control group and experiment group in the first eight days. However, their total remediation was higher, with 56% of their population remediating after 24 days, possibly due to the visibility of the Heartbleed vulnerability.

Compared to security notification studies in other domains, our notifications resulted in similar or lower levels of remediation, despite sharing largely the same methodology and reachability limitations. We were outperformed by the notifications for malware infected sites sent by Vasek and Moore [36] and Li et al. [25], who observed 10-30% of the notified population remediated over the control group, and overall remediation rates were approximately 60% for notified sites. However, our results were similar to studies on vulnerable Git domains, WordPress servers [34], misconfigured IPv6 firewalls, DDos amplifiers, and ICS services [25], which observed differences $<$10% and overall remediation levels of 30% or less.

These results suggest that site owners are less responsive to notifications about HTTPS misconfiguration issues than being infected with malware, or extremely public vulnerabilities like Heartbleed. As we discuss in section 8.3.1, browsers and others could close this gap by applying more incentives for remediation.

8.1.2 Possible barriers to remediation

Based on results from our survey and prior work, we discuss what may have limited the effectiveness of our notifications.

- **Outdated ciphersuites/TLS version:** Some site owners may have wanted to maintain backwards compatibility with older clients. This concern came up in our survey, as well as prior work by Krombholz et al. [22]. Other site owners may have been unable to fix these due to dependencies on a hosting provider, CDN, or other third parties.
- **Certificate expiring soon:** Most sites already seemed to have mechanisms to renew certificates on-time; 85-90% of control sites remediated in the first two weeks. Multiple site owners reported this as a false positive in the survey.
- **Incomplete certificate chain:** Site owners may have ignored this issue because it does not affect most browser users. Except for Firefox, current browsers can automatically fetch intermediate certificates, and Firefox can cache intermediates from other sites, preventing a warning from appearing.

8.2 Lessons on Security Notifications

Despite the limited impact of our notifications on HTTPS misconfigurations, we were able to test several hypotheses about notification methods in general.

First, we found that variations on how we framed the misconfiguration's impact had little effect on remediation rates. Messages highlighting a misconfiguration's impact on users resulted in nearly equivalent remediation rates as messages that focused on the technical details of a misconfiguration. The subject line (general vs. specific) also did not result in any noticeable differences. It appears that these different persuasion tactics do not have an effect on decision making.

Second, we did not find a significant difference in remediation rates between users notified in English versus their native language (both notified via Search Console), suggesting that site owners are comfortable receiving notifications in English. This result contrasts with prior work by Li et al. [24], which found that translated notifications resulted in lower remediation rates, because recipients were likely suspicious of translated messages arriving from US universities. We believe that Search Console messages did not suffer from this effect because users expect messages to be translated, as Google services are available in non-English languages.

Third, we found that notifications sent via Google Search Console were only slightly more effective than messages sent via WHOIS contacts, as used in prior studies [6, 7, 12, 24, 34–36]. We did not detect a significant difference between the channels using the log rank test, but the regression analysis indicated that Search Console messages were marginally more effective. While survey recipients reported trusting Google because of its reputation and because they opted into Search Console messages, this trust did not convert into substantial improvements. This result supports the findings of Cetin et al. [7], who were also unable to find significant differences in remediation between notifications sent by an independent researcher, a university, and an anti-malware organization.

8.3 Recommendations

8.3.1 Combine awareness and incentives to encourage higher remediation of HTTPS issues

Based on our results, it appears that the best strategy for maximizing the remediation of HTTPS misconfigurations and certificate issues is a combination of early public outreach by members of the HTTPS ecosystem, and eventual deployment of browser UI changes to prevent users from accessing misconfigured websites. We observed high baseline remediation levels for the two misconfigurations where there was either an existing outreach effort or impending browser error pages (the Symantec certificate distrust, and notices from CAs about expiring certs).

For other misconfigurations, our notifications had a limited effect, suggesting that while they may bring awareness to site owners, site owners do not feel incentivized to take action. Thus, on their own, these targeted notifications are not sufficient to meaningfully move the HTTPS ecosystem towards a more secure state. However, combined with proper incentives (e.g., browser UI changes to protect users from security problems), these notifications may play a role in the early outreach efforts that can inform webmasters of security issues and the consequences for their site and their users.

8.3.2 Write best effort notification messages

We believe that security notifications are effective as long as they clearly identify the security issue, why it is important, and provide resources to help remediate the issue. As long as these parts of the message are comprehensive, the particular way the security issue is framed in terms of user impact and subject line will not affect the notification's effectiveness.

We also observed that messages do not necessarily need to be translated in order to be effective. This might be due to the English language bias in programming and IT; website admins are likely to have a working proficiency in English so that they can understand documentation and code. While we should strive to internationalize messages and outreach to be more inclusive, for urgent, large-scale, and sensitive security issues, or for small teams of independent security researchers, the cost of translating messages (e.g., personnel, money, time) may outweigh the benefits.

8.3.3 Create open notification channels for researchers

If contacts like email addresses are removed from WHOIS registries (due to GDPR [28]), there should remain alternative methods of reaching site owners that respect their privacy. In theory, CERT organizations could serve as another channel for reaching system administrators. However, Li et al. [24] found that many CERTs were ineffective or did not forward security notifications sent to them.

Another possibility is through opt-in services where website administrators can register to receive (perhaps vetted) security notices. While Google Search Console serves such a purpose, it ultimately lacks comprehensive coverage of websites and access for external researchers. If hosting providers or ISPs explicitly offered such services to their customers, with an avenue for security researchers to report issues, more sites may be reachable.

9 Conclusion

In this study, we ran large-scale controlled experiments with security notifications, in an effort to improve on existing approaches for encouraging site owners to fix HTTPS misconfigurations. We found that HTTPS misconfiguration notifications have a small but statistically significant effect on remediation, but are ineffective for pushing a majority of sites to remediate, unless sent in conjunction with large-scale public outreach and user-facing browser warnings. We also found no significant effect on remediation rates from translating the messages or varying the message framing. Our results indicate that the best way to substantially reduce HTTPS misconfigurations is a combination of public outreach, browser UI changes, and targeted security notifications.

Acknowledgements

We would like to thank the website owners who took the time to provide feedback on our notifications. We also thank Chris Thompson, Christine Chen, Christine Geeng, Kiron Lebeck, and Franzi Roesner for providing feedback on an earlier draft of this paper, and Kurt Thomas for assistance in designing the study. We also are grateful that RiskIQ generously provided us free access to their PassiveTotal WHOIS lookup API for this research. This work was supported in part by the National Science Foundation under awards CNS-1518921 and CNS-1513584.

References

[1] Mozilla CA Certificate Store. `https://www.mozilla.org/en-US/about/governance/policies/security-group/certs/`. Accessed June 23, 2018.

[2] 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. In *ACM Conference on Computer and Communications Security (CCS)*, 2017.

[3] D. Akhawe, B. Amann, M. Vallentin, and R. Sommer. Here's My Cert, So Trust Me, Maybe?: Understanding TLS Errors on the Web. In *International World Wide Web Conference (WWW)*, 2013.

[4] J. M. Bland and D. G. Altman. The logrank test. *British Medical Journal*, May 2004.

[5] N. E. Breslow. Analysis of Survival Data under the Proportional Hazards Model. *International Statistical Review / Revue Internationale de Statistique*, 43(1):45–57, 1975.

[6] O. Cetin, C. Gañán, M. Korczyński, and M. van Eeten. Make Notifications Great Again: Learning How to Notify in the Age of Large-Scale Vulnerability Scanning. In *Workshop on the Economics of Information Security (WEIS)*, 2017.

[7] O. Cetin, M. H. Jhaveri, C. Gañán, M. van Eeten, and T. Moore. Understanding the Role of Sender Reputation in Abuse Reporting and Cleanup. *Journal of Cybersecurity*, 2(1), 2016.

[8] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. `https://tools.ietf.org/html/rfc5280#section-4.2.2.1`. Accessed June 23, 2018.

[9] B. Derrick and P. White. Comparing two samples from an individual Likert question. *International Journal of Mathematics and Statistics*, 18, 2017.

[10] T. Dierks and E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.2. `https://www.ietf.org/rfc/rfc5246.txt`. Accessed June 23, 2018.

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

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

[13] 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 *ACM Symposium on Information, Computer and Communications Security (AsiaCCS)*, 2014.

[14] A. P. Felt, R. Barnes, A. King, C. Palmer, C. Bentzel, and P. Tabriz. Measuring HTTPS Adoption on the Web. In *USENIX Security Symposium*, 2017.

[15] D. Goodin. Critical crypto bug in OpenSSL opens two-thirds of the Web to eavesdropping. `https://arstechnica.com/information-`

technology/2014/04/critical-crypto-bug-in-openssl-opens-two-thirds-of-the-web-to-eavesdropping/, April 2014.

[16] Google. Google crawlers. https://support.google.com/webmasters/answer/1061943?hl=en. Accessed June 28, 2018.

[17] Google. How Google Search Works. https://www.google.com/search/howsearchworks/crawling-indexing/. Accessed June 22, 2018.

[18] N. Heninger, Z. Durumeric, E. Wustrow, and J. A. Halderman. Mining Your Ps and Qs: Detection of Widespread Weak Keys in Network Devices. In *USENIX Security Symposium*, 2012.

[19] S. Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979.

[20] 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 *Internet Measurement Conference (IMC)*, 2011.

[21] R. Housley, W. Polk, W. Ford, and D. Solo. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. https://tools.ietf.org/html/rfc3280#section-4.1.2.5. Accessed June 23, 2018.

[22] K. Krombholz, W. Mayer, M. Schmiedecker, and E. R. Weippl. "I Have No Idea What I'm Doing" - On the Usability of Deploying HTTPS. In *USENIX Security Symposium*, 2017.

[23] M. Kührer, T. Hupperich, C. Rossow, and T. Holz. Exit from Hell? Reducing the Impact of Amplification DDoS Attacks. In *USENIX Security Symposium*, 2014.

[24] F. Li, Z. Durumeric, J. Czyz, M. Y. Karami, M. Bailey, D. McCoy, S. Savage, and V. Paxson. You've Got Vulnerability: Exploring Effective Vulnerability Notifications. In *USENIX Security Symposium*, 2016.

[25] F. Li, G. Ho, E. Kuan, Y. Niu, L. Ballard, K. Thomas, E. Bursztein, and V. Paxson. Remedying Web Hijacking: Notification Effectiveness and Webmaster Comprehension. In *International World Wide Web Conference (WWW)*, 2016.

[26] B. Möller, T. Duong, and K. Kotowicz. This POODLE bites: exploiting the SSL 3.0 fallback. 2014.

[27] Mozilla. Security/Server Side TLS. https://wiki.mozilla.org/Security/Server_Side_TLS#Modern_compatibility, 2018. Accessed June 18, 2018.

[28] D. Oberhaus. WHATIS Going to Happen With WHOIS? https://motherboard.vice.com/en_us/article/vbpgga/whois-gdpr-europe-icann-registrar, February 2018.

[29] D. O'Brien, R. Sleevi, A. Whalley, and C. Security. Chrome's Plan to Distrust Symantec Certificates. https://security.googleblog.com/2017/09/chromes-plan-to-distrust-symantec.html, September 2017. Accessed June 22, 2018.

[30] C. Palmer and R. Sleevi. Gradually Sunsetting SHA-1. https://blog.chromium.org/2014/09/gradually-sunsetting-sha-1.html, 2014. Accessed June 23, 2018.

[31] RiskIQ. WHOIS API. https://api.passivetotal.org/api/docs/#api-WHOIS. Accessed June 27, 2018.

[32] J. Saldaña. *The Coding Manual for Qualitative Researchers*, pages 87–90. SAGE Publications, Incorporated, 2009.

[33] Y. Sheffer, R. Holz, and P. Saint-Andre. Summarizing Known Attacks on Transport Layer Security (TLS) and Datagram TLS (DTLS). *IETF*, 2 2015.

[34] B. Stock, G. Pellegrino, F. Li, M. Backes, and C. Rossow. Didn't You Hear Me? Towards More Successful Web Vulnerability Notifications. In *Network and Distributed System Security Symposium (NDSS)*, 2017.

[35] B. Stock, G. Pellegrino, C. Rossow, M. Johns, and M. Backes. Hey, You Have a Problem: On the Feasibility of Large-Scale Web Vulnerability Notification. In *USENIX Security Symposium*, 2016.

[36] M. Vasek and T. Moore. Do Malware Reports Expedite Cleanup? An Experimental Study. In *Workshop on Cyber Security Experimentation and Test (CSET)*, 2012.

[37] K. Wilson. Distrust of Symantec TLS Certificates. https://blog.mozilla.org/security/2018/03/12/distrust-symantec-tls-certificates/, March 2018. Accessed June 22, 2018.

[38] K. L. Wuensch. The Common Language Effect Size Statistic. http://core.ecu.edu/psyc/wuenschk/docs30/CL.pdf, 2015. Accessed June 28, 2018.

Appendices

A AEAD Ciphersuites

This is a list of TLS ciphersuites that support Authenticated Encryption with Associated Data (AEAD), which we considered as not outdated in our notifications. In our notifications, we recommended that sites use these ciphersuites.

- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (c0,2b)
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (c0,2c)
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (c0,2f)
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (c0,30)
- TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305 (cc,a8)
- TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305 (cc,a9)

B Notification Message Content

We include the two variants of the incomplete chain notification message, sent to the WHOIS email addresses, to illustrate the differences between the user focus and technical focus conditions.

B.1 Incomplete Chain, WHOIS, User focus

To owner of *${site_url}*,
We are a team of computer security researchers at the *${university name}* studying HTTPS configurations on websites. We recently detected that your server is not serving all of the intermediate certificates in the TLS certificate chain for *${site_url}*. Due to this configuration issue, Mozilla Firefox and Chrome for Android (versions older than v58) are currently blocking users from accessing *${site_url}* with a security warning message.
If you have already corrected this problem, you can ignore this message. Thank you for making your website safer for your users.
Because of this misconfiguration, Firefox and Chrome for Android will be unable to verify that the connection to your server is secure. All users of these browsers will see a full-screen security warning, and will be unable to access your site. This is done to protect users browsing data, such as passwords, page content, and form data, from being intercepted or tampered with by a third party.
Here's how to fix this problem:

1. *Configure your server to provide all intermediate certificates*
   When obtaining your SSL certificates, your certificate authority should provide all of the necessary intermediate certificates. Ensure that your server serves all intermediate certificates to clients.
   Here is some documentation on how to install intermediate certificates for the most common types of web servers:
   - Apache
   - IIS
   - nginx
2. *If you do not run your own server, contact your hosting provider to resolve this issue.*
   Let your hosting provider know that your users are seeing warnings in their browser because your sites servers are providing incomplete TLS certificate chains.

For more information about these security notifications, please visit our website at: *${supplement_url}*

Was this message helpful? Please take our survey: *${survey url}*

B.2 Incomplete Chain, WHOIS, Technical focus

To the owner of *${site_url}*,
We are a team of computer security researchers at the *${university name}* studying HTTPS configurations on websites. We recently detected that your server is not serving all of the intermediate certificates in the TLS certificate chain for ${site_url}. Due to this configuration issue, Mozilla Firefox and Chrome for Android (versions older than v58) are currently blocking users from accessing ${site_url} with a security warning message.
If you have already corrected this problem, you can ignore this message. Thank you for making your website safer for your users.
Intermediate certificates are used to create a chain of trust between root certificates that are trusted by the browser, and leaf certificates issued for HTTPS sites. Because your site is missing one or more intermediate certificates, web browsers may be unable to validate a chain from your certificate to a trusted root.
While your website may function properly on other browsers that cache intermediate certificates, Mozilla Firefox and Chrome for Android dont support these features. Users of those browsers will be shown full-screen security warnings, and will be unable to access your site.
Here's how to fix this problem:

1. *Configure your server to provide all intermediate certificates*
   When obtaining your SSL certificates, your certificate authority should provide all of the necessary intermediate certificates. Ensure that your server serves all intermediate certificates to clients.
   Here is some documentation on how to install intermediate certificates for the most common types of web servers:

- Apache
- IIS
- nginx

2. *If you do not run your own server, contact your hosting provider to resolve this issue.*
   Let your hosting provider know that your users are seeing warnings in their browser because your sites servers are providing incomplete TLS certificate chains.

For more information about these security notifications, please visit our website at: *${supplement_url}*

Was this message helpful? Please take our survey: *${survey url}*

C Survey Questions

1. Is your organization planning on making any changes or fixes after receiving our message? (Yes/No)
2. Was your organization aware of the misconfiguration prior to our message? (Yes/No)
3. How did your organization first become aware of the misconfiguration? (Free response)
4. Did your organization take prior actions to resolve the misconfiguration before our message? (Yes/No)
5. What prior actions did your organization take, if any? (Free response)
6. Was it clear from our message what the misconfiguration is? (Yes/No)
7. If not, is there a way we can improve our explanation? (Free Response)
8. Based on our message, how serious does the misconfiguration seem? (Likert Scale)
9. Was it clear from our message how to address the issue? (Yes/No)
10. Did you do further research after seeing our message to better understand the misconfiguration? (Yes/No)
11. How trustworthy was our message? (Likert scale)
12. How acceptable was it for us to detect the misconfiguration and notify you about the problem? (Likert scale)
13. How interested would your organization be in receiving similar security and misconfiguration notifications in the future? (Likert scale)
14. Is there any way we can improve our notifications, or anything else you wanted to tell us? (Free response)

D Statistical Analysis Results

D.1 Pairwise Log Rank Test Results

Variable	Comparison	Misconfiguration Type	n_a	n_b	Test Statistic	p-value	Significant?
Sender	Google vs. Control	Outdated Ciphersuite	4000	1248	5.38	0.0203	
Sender	Google vs. Control	Outdated TLS Version	4000	1108	18.8	**<0.0001**	*
Sender	Google vs. Control	Incomplete Chain	4000	1180	23.9	**<0.0001**	*
Sender	Google vs. Control	Cert Expires Soon	2000	1222	0.491	0.4834	
Sender	WHOIS vs. Control	Outdated Ciphersuite	1964	1590	3.85	0.0498	
Sender	WHOIS vs. Control	Incomplete Chain	1953	1537	1.86	0.1724	
Sender	WHOIS vs. Control	Cert Expires Soon	973	1638	2.71	0.0996	
Sender	WHOIS vs. Control	Outdated TLS Version	1957	1613	1.16	0.2820	
Sender	Google vs. WHOIS (A)	Outdated Ciphersuite	630	216	0.281	0.5963	
Sender	Google vs. WHOIS (A)	Incomplete Chain	573	205	0.0611	0.8048	
Sender	Google vs. WHOIS (A)	Outdated TLS Version	606	119	0.488	0.4850	
Sender	Google vs. WHOIS (B)	Outdated Ciphersuite	630	235	0.0587	0.8085	
Sender	Google vs. WHOIS (B)	Incomplete Chain	569	202	1.02	0.3130	
Sender	Google vs. WHOIS (B)	Outdated TLS Version	618	122	2.59	0.1073	
Sender	Google vs. WHOIS	Cert Expires Soon	617	227	0.254	0.6142	
Population	In SC vs. Not in SC (A)	Outdated Ciphersuite	216	764	0.964	0.3262	
Population	In SC vs. Not in SC (A)	Incomplete Chain	205	776	8.46	**0.0036**	*
Population	In SC vs. Not in SC (A)	Outdated TLS Version	119	863	3.85	0.0497	
Population	In SC vs. Not in SC (B)	Outdated Ciphersuite	235	749	0.202	0.6533	
Population	In SC vs. Not in SC (B)	Incomplete Chain	202	770	10.0	**0.0015**	*
Population	In SC vs. Not in SC (B)	Outdated TLS Version	122	853	0.00864	0.9260	
Population	In SC vs. Not in SC	Cert Expires Soon	227	746	13.7	**0.0002**	*
Subject	A vs. B (Google, English)	Outdated Ciphersuite	1000	1000	0.176	0.6751	
Subject	A vs. B (Google, English)	Outdated TLS Version	1000	1000	0.0211	0.8844	
Subject	A vs. B (Google, Translated)	Outdated Ciphersuite	1000	1000	1.04	0.3068	
Subject	A vs. B (Google, Translated)	Outdated TLS Version	1000	1000	0.164	0.6851	
Subject	A vs. B (WHOIS)	Outdated Ciphersuite	980	984	0.838	0.3600	
Subject	A vs. B (WHOIS)	Outdated TLS Version	982	975	2.24	0.1346	
Framing	A vs. B (Google, Translated)	Incomplete Chain	1000	1000	1.02	0.3114	
Framing	A vs. B (Google, English)	Incomplete Chain	1000	1000	0.660	0.4164	
Framing	A vs. B (WHOIS)	Incomplete Chain	981	972	0.650	0.4199	
Language	English vs. Translated (Google, A)	Outdated Ciphersuite	1000	1000	0.0647	0.7992	
Language	English vs. Translated (Google, A)	Incomplete Chain	1000	1000	2.40	0.1212	
Language	English vs. Translated (Google, A)	Outdated TLS Version	1000	1000	0.917	0.3383	
Language	English vs. Translated (Google, B)	Outdated Ciphersuite	1000	1000	0.121	0.7279	
Language	English vs. Translated (Google, B)	Incomplete Chain	1000	1000	0.0760	0.7837	
Language	English vs. Translated (Google, B)	Outdated Ciphersuite	1000	1000	0.171	0.6791	
Language	English vs. Translated (Google)	Cert Expires Soon	1000	1000	0.103	0.7473	

Table 7: Pairwise Log-Rank Test comparisons for notifications sent in March 2018. We determine significance using a p-value threshold of $\alpha = 0.05$, applying the Holm-Bonferroni method for multiple comparison corrections.

D.2 Cox Proportional Hazards Regression Results

Misconfiguration Type	Factor	coef	exp(coef)	se(coef)	z	p	lower 0.95	upper 0.95
Outdated Ciphersuite	Sender: University	0.4498	1.5680	0.1468	3.0649	**0.0022****	0.1622	0.7374
Outdated Ciphersuite	Sender: Google	0.3413	1.4067	0.1261	2.7070	**0.0068****	0.0942	0.5884
Outdated Ciphersuite	In Search Console	0.2502	1.2842	0.1457	1.7169	0.0860	-0.0354	0.5357
Outdated Ciphersuite	In Alexa Top 1m	-0.1970	0.8212	0.3063	-0.6433	0.5200	-0.7974	0.4033
Outdated Ciphersuite	In Cisco Top 1m	-0.5909	0.5538	0.1716	-3.4429	**0.0006*****	-0.9274	-0.2545
Outdated TLS Version	Sender: University	0.1970	1.2178	0.1036	1.9018	0.0572	-0.0060	0.4000
Outdated TLS Version	Sender: Google	0.3970	1.4873	0.0837	4.7405	<**0.0001*****	0.2328	0.5611
Outdated TLS Version	In Search Console	0.3140	1.3690	0.1063	2.9556	**0.0031****	0.1058	0.5223
Outdated TLS Version	In Alexa Top 1m	-0.1322	0.8762	0.1869	-0.7072	0.4794	-0.4986	0.2342
Outdated TLS Version	In Cisco Top 1m	-0.1409	0.8686	0.1757	-0.8021	0.4225	-0.4853	0.2035
Incomplete Chain	Sender: University	0.2225	1.2492	0.0707	3.1466	**0.0017****	0.0839	0.3610
Incomplete Chain	Sender: Google	0.3025	1.3532	0.0580	5.2158	<**0.0001*****	0.1888	0.4161
Incomplete Chain	In Search Console	0.4124	1.5104	0.0709	5.8204	<**0.0001*****	0.2735	0.5512
Incomplete Chain	In Alexa Top 1m	-0.2206	0.8021	0.1267	-1.7406	0.0818	-0.4689	0.0278
Incomplete Chain	In Cisco Top 1m	-0.4482	0.6388	0.1419	-3.1587	**0.0016****	-0.7263	-0.1701
Cert Expires Soon	Sender: University	-0.0103	0.9897	0.0440	-0.2351	0.8141	-0.0966	0.0759
Cert Expires Soon	Sender: Google	0.0226	1.0229	0.0360	0.6287	0.5296	-0.0480	0.0932
Cert Expires Soon	In Search Console	0.1952	1.2156	0.0414	4.7173	<**0.0001*****	0.1141	0.2764
Cert Expires Soon	In Alexa Top 1m	-0.1932	0.8243	0.1026	-1.8831	0.0597	-0.3942	0.0079
Cert Expires Soon	In Cisco Top 1m	-0.3027	0.7388	0.1503	-2.0137	**0.0440***	-0.5973	-0.0081

Significance Codes: 0 *** 0.001 ** 0.01 * 0.05 ‘ ’ 1

Table 8: Cox-Proportional Hazards Regression on remediation data from sites notified in March 2018. exp(coef) indicates the hazard ratio between the particular factor and the baseline condition. For example, Outdated Ciphersuite misconfigurations notified by Google are 1.41 times more likely to remediate than the control.