On the Usability of HTTPS Deployment

Matthew Bernhard
University of Michigan
matber@umich.edu

Jonathan Sharman
Rice University
jonathan.p.sharman@rice.edu

Claudia Ziegler Acemyan
Rice University
claudiaz@rice.edu

Philip Kortum
Rice University
pkortum@rice.edu

Dan S. Wallach
Rice University
dwallach@rice.edu

J. Alex Halderman
University of Michigan
jhalderm@umich.edu

ABSTRACT

HTTPS and TLS are the backbone of Internet security, however setting up web servers to run these protocols is a notoriously difficult process. In this paper, we perform two live subjects usability studies on the deployment of HTTPS in a real-world setting. Study 1 is a within subjects comparison between traditional HTTPS configuration (purchasing a certificate and installing it on a server) and Let's Encrypt, which automates much of the process. Study 2 is a between subjects study looking at the same two systems, examining why users encounter usability issues. Overall we confirm past results that HTTPS is difficult to deploy, and we find some evidence that suggests Let's Encrypt is an easier, more efficient method for deploying HTTPS.

ACM Reference Format:
Matthew Bernhard, Jonathan Sharman, Claudia Ziegler Acemyan, Philip Kortum, Dan S. Wallach, and J. Alex Halderman . 2019. On the Usability of HTTPS Deployment. In *CHI Conference on Human Factors in Computing Systems Proceedings (CHI 2019), May 4–9, 2019, Glasgow, Scotland UK*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3290605.3300540

1 INTRODUCTION

The modern Internet is largely secured via the HTTPS protocol, which relies on Transport Layer Security (TLS). Google Chrome recently integrated a feature that automatically treats websites not served over HTTPS as "not secure" [25]. However, TLS is known as a notoriously difficult protocol to set up and deploy successfully. Certificates are often expensive and difficult to acquire. Misconfiguration and the difficulty of modifying server configurations to mitigate known attacks have resulted in numerous web vulnerabilities, and misconfiguration exacerbates attacks like FREAK, DROWN, and POODLE [11, 12, 19]. Recent efforts like Let's Encrypt have attempted to remedy this problem by making TLS deployment automated and free, and a majority of websites are now using the HTTPS protocol [14].

In this paper, we study the usability of deploying TLS, both manually and using Let's Encrypt. We study a diverse set of participants purchasing certificates from real certificate authorities, configuring an Apache2 webserver, and contrast the usability of this method of TLS deployment against that of Let's Encrypt and the EFF's certbot. We verify the findings of Krombholz et al. [17], and expand it with a more diverse sample and more real-world testing conditions.

Overall, we find that HTTPS deployment is in fact very difficult, even for users with technical expertise. Neither manual configuration nor Let's Encrypt resulted in all of our participants successfully deploying HTTPS. However, Let's Encrypt did see higher rates of success than manual configuration and lower time on task, indicating that Let's Encrypt is a more usable way to deploy HTTPS. We believe these findings provide limited support for the argument that Let's Encrypt makes the Internet a safer place.

2 BACKGROUND

Usability

Our study evaluates usability based on the ISO 9241-11 standard, which provides three categories for evaluation: effectiveness, efficiency, and satisfaction [15]. For efficiency, we measure the amount of time it takes for a user to completely deploy TLS. For satisfaction, we rely on the system usability score (SUS), which was developed by Brooke [9], and evaluated by Bangor et al. in [7, 8]. For effectiveness, we rely on SSLLabs' SSL Server Test [22], an external evaluation service that provides a letter grade score determining how secure a given website is. What makes a TLS connection "secure", and how SSLLabs determines this and provides a score, is explained further in Section 2.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
CHI 2019, May 4–9, 2019, Glasgow, Scotland UK
© 2019 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-5970-2/19/05.
https://doi.org/10.1145/3290605.3300540

Table 1: SSL Labs Grades Mapping—a mapping of SSL Labs letter grades to the corresponding numerical values

Letter Grade	Numeric Grade	Letter Grade	Numeric Grade
A+	100	C-	46.67
A	93.33	D+	40
A-	86.67	D	33.33
B+	80	D-	26.67
B	73.33	E+	20
B-	66.67	E	13.33
C+	60	E-	6.67
C	53.33	F	0

Enabling HTTPS

In order to enable HTTPS, site operators must first acquire a trusted certificate from one of several certificate authorities (CAs), and then install the certificate on their web server. Most certificate authorities require that a site operator submit a self-generate certificate signing request (CSR), which contains information about the website and site operator necessary for creating a trusted end-entity certificate. After submitting a CSR, site operators can download a certificate bundle that they can install onto their web server. On Apache2, the web server that we use in this study, this requires updating web server configuration to use the proper certificate file, private key file, and certificate chain file generated by the certificate authority. Users can also change the server configuration to support different variants of the SSL/TLS protocol, which trades compatibility for security, as older version of the protocol have known vulnerabilities like FREAK, DROWN, and POODLE, to name a few [11, 12, 19].

This is not the workflow for all CAs. Let's Encrypt, a free certificate authority that we investigate in this study, does not require site operators to submit a CSR or set up their web server manually. Instead services like the Electronic Frontier Foundation's `certbot` [1] program interact with the Let's Encrypt CA and generate a certificate automatically, as well as updating the web server configuration to support HTTPS with the newly generated certificate.

Evaluating Security

Qualys SSL Labs [22] provides a web service where users can type in the name of a website, and then receive a detailed break down of the HTTPS configuration of the server hosting that website, along with a letter grade assessing the overall security provided by the site. Table 1 provides a mapping of letter grades to numerical score, out of 100, which we use throughout the paper in discussing SSL Labs results as well as user sentiment, e.g. likelihood to recommend and overall ease-of-use.

3 METHODOLOGY

Our experiment ran in two studies. The first was a within-subjects comparison between Let's Encrypt and manual certificate acquisition and server configuration. The second study was between-subjects, studying the specific error modes of using each method from Study 1. We obtained IRB approval for both studies.

Participants

Subjects for both studies were recruited via Craigslist and Facebook ads or via connections with the researchers. Each participant was paid $50 for participating in this study. We attempted to recruit a diverse sample of participants with some UNIX knowledge, ranging from hobbyists to system administrators. All participants were recruited from the Houston area and self-reported normal or corrected-to-normal vision. Recruitment for this study under these constraints was very difficult, which greatly limited the number of participants we could recruit. We further discuss the limitations imposed by our recruitment later in Section 5.

Study 1. We recruited nine participants in total for Study 1, eight of which are included in the results. (One participant had to be excluded from the study due to a methodological error during the experiment.) Seven of the participants spoke English natively, and the remaining participant was a native speaker of Telugu. The mean age was 26.4 years, with a median of 23 and a range of 19 to 41. Seven participants were male, and one was female. Participants were Asian (7, 87.5%) and African American (1, 12.5%). The highest education levels of the participants were bachelor's degree or equivalent (4, 50%), some college or associate's degree (2, 25%), high school or G.E.D. (1, 12.5%), and postgraduate degree (1, 12.5%). Participants self-reported their computer expertise on a scale from 1 to 10, with a mean of 6.1, median of 7, and range of (1, 10).

Study 2. We recruited ten participants in Study 2, assigning five to use Let's Encrypt and five to use manual HTTPS deployment methods. All participants spoke English natively. The mean age in the Let's Encrypt group was 40.4, with a median of 37 and a range of 28 to 63. The mean age in the manual group was 41.4 years, with a median of 42 and a range of 29 to 64. All participants in both groups were male. The Let's Encrypt participants' ethnicities were Caucasian (3, 60%) and African American (2, 40%), and those of the manual participants were Caucasian (2, 40%), African American (1, 20%), Asian (1, 20%), and Mexican American/Chicano (1, 20%). The highest education levels for Let's Encrypt participants were bachelor's degree (3, 60%), postgraduate degree (1, 20%), and some college or associate's degree (1, 20%); manual participants had a bachelor's or equivalent (2, 40%),

postgraduate degree (2, 40%), and some college or associate's degree (1, 20%). Let's Encrypt participants' self-reported computer expertise scores on a scale from 1 to 10 had a mean and median of 8 and a range of (7, 10). For manual participants, the expertise scores had a mean of 8.2, median of 8, and range of (7, 10).

Design

We measured efficiency as the total time to complete a task. Subjects were limited to two hours to finish the task.

Study 1. Each participant attempted to deploy HTTPS on an Apache2 web server using both Let's Encrypt and manual acquisition, deployment, and configuration. The order of the two tasks was assigned at random for each participant, to control for possible ordering effects. In this study, we measured effectiveness as either success or failure. The participant succeeded if and only if the test server was accessible via HTTPS after they completed the task.

Study 2. Each participant was assigned at random to attempt to deploy HTTPS using either Let's Encrypt or manual deployment. In this study, in addition to success or failure as a measure of effectiveness, we also recorded the Qualys SSL Labs score for each server configuration. SSL Labs provides a letter grade from F to A+ indicating the level of security of a server's HTTPS configuration. Since SSL Labs does not provide a score for a server that cannot communicate using HTTPS, we considered such servers to have a score of F for the purpose of analysis.

To assess participants' understanding of the task and their satisfaction with both methods, we issued a System Usability Score (SUS) survey after each task [9], as well as asking participants for comments about the system. We consider shorter times on task, higher SSL Labs scores, high SUS scores, and positive comments from participants to indicate that a system is more usable.

Procedures

In both studies, we obtained informed consent from each participant before beginning and then read them instructions for their first task.

Participants were read instructions about both studies from a script (see Appendix A). For Study 1, we instructed participants to acquire a TLS certificate and then deploy it on the provided server, using either Let's Encrypt or manual configuration. After completing the task for one method, participants would fill out a questionnaire and then repeat the task with the other method. In Study 2, participants were similarly instructed to acquire a certificate using one method or the other. The choice or ordering of methods for each participant in each Study was determined by alternating between assignments while attempting to keep a uniform distribution of demographics between both experimental groups.

Participants were provided with an information sheet containing login credentials, certificate authority information, and billing information to enable them to purchase certificates or fill out Let's Encrypt data (see Appendix B for an example). At the beginning of each session, participants were given a laptop with a blank Safari tab open and a terminal that was already SSH'd into the remote server, to avoid added complexity in instructions. Participants were permitted to use any Internet resources they could find, including but not limited to tutorials and instruction sites from CAs.

Manual Configuration. For the manual portions of studies 1 and 2, participants were expected to navigate to the CA's website and purchase a certificate. We chose Namecheap [2] as our CA, as its certificates are relatively inexpensive and it is a fairly commonly used service. Once purchased, the CA asked for a certificate signing request, which is generated via OpenSSL [3] on the remote server. Participants would then download the certificate and upload it to the remote server. After upload, participants would configure a default Apache2 configuration to use the certificate.

Let's Encrypt. For the Let's Encrypt portions of both studies, participants were expected to navigate to Let's Encrypt's website and download certbot to the remote server. Participants then followed the prompts on the remote server to acquire a certificate, with the server configuration being done automatically by certbot.

Study 1. If the participant spent more than ten minutes not making any progress (e.g. staring at the same web page), the experimenter stopped them.[1] After attempting each task, each participant answered a set of survey questions about the subjective usability of the system they just used; however, these were ultimately not used due to an incident in which some of the survey results from Study 1 were lost. After finishing with both tasks, participants filled out a survey with demographic and TLS background information.

Study 2. In this study, participants were allowed to keep trying to complete the task even if they became stuck, as long as they remained within the three-hour time limit. As in Study 1, after attempting the task participants filled out a survey, providing a SUS score of the system to which they were assigned, along with demographic and background information.

Finally, in both studies, participants were debriefed, compensated, and thanked for participating in the study.

[1] Ten minutes was chosen following precedent of related work (e.g. [16]. However, we ultimately decided that this limitation artificially hindered our participants and removed it for our second study.

Table 2: Results for the Study 1—Time on task, and success for each subject in study 1.

Subject	Manual Time (s)	Manual Success	LE Time (s)	LE Success
1	981	N	579	Y
2	5406	N	520	Y
3	5820	N	420	Y
4	2220	N	3180	N
5	3060	N	2820	N
7	300	N	1140	N
8	3720	N	3240	N
9	6420	N	300	Y
Avg.	3491.0	0	1525.0	0.5
Std. Dev.	2268.0	0	1316.5	0.5

Materials

Both studies of the study required similar work flows for the participants. We gave each participant a unique GMail account. For participants in Study 1 or the manual configuration of Study 2, we also provided a Visa gift card with enough value to purchase a certificate. Finally, participants were sat at a 2015 MacBook Pro 15-inch laptop, operating in guest user mode, to complete the study. For the remote server, we allocated an Amazon EC2 instance running Ubuntu 16.04, with Apache2 preinstalled. We chose Apache2 as it is the most common server deployed on Linux [20] and has dominated market share over the past two decades. Thus, Apache2 provides a baseline for server configuration experience.

We obtained demographic information and satisfaction data via SurveyMonkey [4] forms.

4 RESULTS

This section summarizes the quantitative and qualitative findings for both Study 1 and Study 2. In Study 1 (within-subjects), we ran a paired t-test for each measured quantity, including measures of effectiveness and efficiency. In Study 2 (between-subjects), we used an unpaired Welch's t-test and measured satisfaction, in addition to effectiveness and efficiency. Results differed between Study 1 and Study 2. In Study 1, we observed a reliable effect of deployment method on effectiveness but not on efficiency. In Study 2, we observed a reliable effect of deployment method on efficiency but not on effectiveness or satisfaction.

Study 1

Results for Study 1 are summarized in Table 2.

Effectiveness. To evaluate effectiveness, we treat a passing server configuration as a 1 and a failing configuration as a 0. In Study 1, no participants were able to successfully deploy HTTPS using the manual deployment method, $\bar{x} = 0$, $s = 0$. Four participants were able to deploy HTTPS using Let's Encrypt, $\bar{x} = 0.500$, $s = 0.535$. There was a reliable effect of deployment method on task completion, t = -2.646, 7 degrees of freedom, p = 0.033. The Cohen's d effect size was -0.935, indicating a large effect.

Efficiency. For the efficiency of manual deployment, we observed $\bar{x} = 3491$ s, $s = 2267.999$ s, and range = (300 s, 6420 s). For Let's Encrypt, $\bar{x} = 1525$ s, $s = 1316.494$ s, and range = (300 s, 3240 s). There was no reliable evidence that time on task differed between manual deployment and Let's Encrypt, t = 1.874, 7 degrees of freedom, p = 0.103, though we note that this may have been due to the restrictive time limit we placed on participants..

Table 3: Results for Study 2 Manual configuration

Subject	Time (s)	SSLLabs Grade	SUS	LR	UF
102	8116	0 (F)	12.5	2	33.33 (D)
104	6881	73.33 (B)	52.5	6	46.67 (C-)
105	4256	86.67 (A-)	35.0	4	66.67 (B-)
108	10127	86.67 (A-)	10.0	1	26.67 (D-)
109	7221	0 (F)	5.0	2	26.67 (D-)
Avg.	7320	49.33 (C-)	23.0	3	40.00 (D+)

LR—likely to recommend UF—user friendliness

Table 4: Results for Study 2 Let's Encrypt configuration

Subject	Time (s)	SSLLabs Grade	SUS	LR	UF
101	2874	100.0 (A+)	45.0	5	73.33 (B)
103	2048	93.33 (A)	92.5	10	93.33 (A)
106	488	0 (F)	15.0	1	0 (F)
107	8413	0 (F)	67.5	8	66.67 (B-)
110	769	93.33 (A)	97.5	10	93.33 (A)
Avg.	2918	57.33 (C)	63.5	6.8	65.33 (C+)

LR—likely to recommend UF—user friendliness

Study 2

Results for Study 2 are summarized in Tables 3 and 4.

Effectiveness. Completion rates were identical between both systems in Study 2 with three participants completing the task and two failing for each system, $\bar{x} = 0.6$, $s = 0.548$. Thus, in contrast to the results of Study 1, there was no reliable effect of the system used on effectiveness as measured by success or failure, $t = 0$, 8 degrees of freedom, $p = 1$.

In Study 2, we recorded Qualys SSL Labs grades – in addition to binary pass/fail data – to obtain more fine-grained

information about how secure participants' solutions were. It is possible for two servers that each serve HTTPS to have different degrees of security, depending on the specifics of each server's configuration. SSL Labs provides a letter grade for a given server, deducting points for configurations that can result in security vulnerabilities and awarding bonus points for especially secure server practices. For analysis, we converted the provided letter grades into a numerical grade from 0 to 100, 0 corresponding to a grade of "F" and 100 to a grade of "A+".

For numeric grades when using manual deployment, $\bar{x}$ = 49.33, s = 45.36, and range = (0, 86.67). For Let's Encrypt, $\bar{x}$ = 57.33, s = 52.41, and range = (0, 100). While the sample mean grade was slightly lower for manual compared to Let's Encrypt, the effect was not reliable, $t = -0.258$, 7.8389 degrees of freedom, $p = 0.803$.

Based on our experience in this Study, grades are essentially bimodal. Even the default Apache configuration with no manual tuning obtained no less than a B according to SSL Labs. Therefore, while additional tuning or the use of automated configuration tools may improve the server's grade slightly, these benefits are marginal compared to passing or failing.

Efficiency. For participants assigned to manual deployment, $\bar{x}$ = 7320s, s = 2127s, and range = (4256 s, 10,127 s). For participants assigned to Let's Encrypt, $\bar{x}$ = 2918s, s = 3220s, and range = (488 s, 8413 s). There was a reliable effect of deployment system on efficiency, i.e. participants using Let's Encrypt took less time compared to participants using manual deployment to either complete the task or determine that they could not complete the task, t = 2.550, 6.933 degrees of freedom, p = 0.0384. Cohen's d for this property was 1.613, indicating that the effect of deployment method on efficiency is large.

These results differ notably from the efficiency results in Study 1. We discuss possible explanations for this discrepancy in Section 5.

Satisfaction. To assess satisfaction, we report quantitative and qualitative measures, namely SUS scores and comments from participants.

Quantitative Measures. Each participant in Study 2 completed a SUS survey about the assigned HTTPS deployment method, providing a numerical score from 0 (worst) to 100 (best). For manual deployment SUS scores, $\bar{x}$ = 23.0, s = 20.109, and range = (5.0, 52.5). For Let's Encrypt SUS scores, $\bar{x}$ = 63.5, s = 34.3, and range = (15, 97.5). At a 95% confidence interval, there was no reliable effect of deployment method on satisfaction, t = -2.277, 6.459 degrees of freedom, p = 0.060. Cohen's d was -1.440, indicating a large effect size.

In addition to the standard SUS questions, we asked each participant to provide quantitative ratings of his assigned deployment system, viz. a letter grade of the overall user-friendliness of the system, how likely she would be to recommend the method to a friend from 1-10, and how confident she was in the security of the system she had just set up.

Each participant provided a letter grade representing the user-friendliness of the assigned deployment method. We then converted these scores to a 100-point scale, using the same mapping we used to convert SSL Labs scores.

We did not find reliable differences in the other satisfaction metrics we examined. For manual user-friendliness, $\bar{x}$ = 40.00, s = 16.997, range (26.67, 66.67). Let's Encrypt user-friendless was $\bar{x}$ = 65.33, s = 38.412, range (0, 93.33). At a 95% confidence interval, there was no reliable effect: Welch Two Sample t-test: t = -1.3484, df = 5.5087, p-value = 0.2303, and Cohen's d was -0.8528179, indicating a large effect size.

For manual recommendation likelihood, $\bar{x}$ = 3, s = 2.000, range (1, 6). Let's Encrypt recommendation likelihood was $\bar{x}$ = 6.8, s = 3.834, range (1, 10). Again, we found no reliable effect, with Welch Two Sample t = -1.9649, df = 6.0268, p-value = 0.09682 and a large effect size from Cohen's d = -1.242733.

Finally, we found no reliable effect on the confidence users had between the two methods. For manual configuration, confidence was $\bar{x}$ = 4, s = 3.082, range (1, 8). For Let's Encrypt, confidence was $\bar{x}$ = 6.2, s = 2.168, range (3, 8). Welch Two Sample t = -1.3055, df = 7.1796, p-value = 0.232, and a large effect size from Cohen's d: -0.8256453.

The lack of reliable effects between the two is possibly due to the small dataset, but could also be due to both methods being somewhat technically intricate and not designed to be user-friendly.

Qualitative Measures. We collected comments from each participant about the HTTPS deployment method to which she was assigned, addressing the satisfaction of using that method.

When asked whether anything about manual HTTPS deployment was frustrating, three of the five participants referred to the domain control validation (DCV) process. Specific problems with the DCV process included difficulty in telling whether the process was underway, long delays between starting and finishing the process, and insufficient feedback from the CA.

Users of both systems cited frustration by unfamiliarity with the process and tools. One user of Let's Encrypt was frustrated that Let's Encrypt did not disable TLS 1.0 and 1.1 by default.

The justifications for confidence ratings were different for each manual deployment participant. One participant rated his confidence in his system as 1 (not at all confident), saying

simply "I didn't complete the assignment". Another participant who gave a rating of 1 listed three concerns, including that the server was running on Apache, that he did not disable HTTP, and that he did not think he had properly secured his private key. A participant who gave a rating of 4 stated "The configuration process did not give me enough information to determine the available security options and their relative strengths." The participant with a rating of 6 simply stated "you can generate csr key using public tool". The most confident participant (8) said that the SSL seemed to be working fine based on the fact that the site could be accessed by HTTPS.

Let's Encrypt users overall rated their confidence in their system higher than users of the manual system, but all still had qualms about the system. Two participants (with ratings 3 and 7) mentioned unfamiliarity/lack of knowledge as reasons for lowering their confidence rating, and one (5) mentioned lack of testing. Another participant (8) wondered "if you get what you pay for", referring to the fact that Let's Encrypt is a free tool that obtains a free certificate. The last participant (8) said he was "very confident, if both obsolete versions of TLS are disabled".

5 DISCUSSION

Discrepancies between Study 1 and Study 2

As reported in Section 4, there were differences in the results between phase 1 and phase 2 of this study. The biggest methodological change between phase 1 and phase 2 was going from a within-subjects to a between-subjects design.

One key methodological difference between the two phases that could partially explain this is that in the first study, we stopped the participants if they failed to make any forward progress after ten minutes (for example, if they just spent ten minutes scrolling through web search results and not working on the task). In phase 2, we allowed participants to keep trying as long as they wanted within the allotted time. Often these subjects would appear to be completely stuck but would eventually figure out how to proceed. Preempting participants reduces both their time-on-task and their success rate, which could partially explain both why manual completion times were relatively slower in the second phase and why manual completion rates were relatively higher.

We also slightly changed the wording of our recruitment criteria. In our recruitment materials for phase 1, we solicited participants with "working knowledge of UNIX-like operating systems and Internet infrastructure". In phase 2, we replaced this wording with "working knowledge of the Unix shell and either experience configuring web servers or the intention to configure a web server in the future". Our intent with this change was to focus our recruitment efforts on the specific skills of our target population, i.e. ability to deploy and configure web servers. This more targeted language may have attracted participants with more relevant technological experience. It is possible that Let's Encrypt is easier to deploy for non-experts but that the gap between Let's Encrypt and manual deployment narrows among more experienced users, but we do not have sufficient data to draw this conclusion.

Difficulties with Recruitment

University students, while readily available to serve as participants, are not necessarily representative of the entire population of people who are likely to set up an externally visible web server. One of the goals of this study was to obtain a more representative sampling of this population. However, we found it extremely difficult to recruit a sufficient number of well qualified participants, which diminishes the statistical power of our results. We suspect that a significant barrier to recruitment was the modest level of compensation we offered for participants' time. Given the high advertised upper bound for the duration of the study (around three hours), the high average level of education, job experience, and income of target participants, and the busy schedule of typical web developers, $50 may simply not have been enough for many would-be participants to justify spending the time to participate.

Conclusions

Taking the results of both phases of the study together, we find limited evidence that Let's Encrypt is more effective at enabling participants to deploy HTTPS, and that it allows them to do so more efficiently. Let's Encrypt obviates several of the common sources of mistakes and frustration during the deployment process, such as uncertainty over the choice of certificate to purchase, confusion surrounding CSRs, and difficulties in the domain validation stage. We believe these advantages are significant enough to recommend Let's Encrypt over manual methods.

However, we did not find conclusive evidence regarding which method is more satisfactory to users, which enables more secure configurations, which system users were more confident in, nor which systems users would recommend. This is likely due to our small sample size, and future work is needed to better understand these features.

6 RELATED WORK

Our work follows a chain of research that investigates the relationship that users have with TLS and HTTPS. Much of this work has focused on understanding and improving SSL browser warnings [5, 6, 23]. Most recently, Reeder et al. extended this work by surveying user reactions to browser warnings in the field [21], and find that many users have different reasons for following browser warnings depending on their individual context.

Other work has focused on why webmasters and site operators make HTTPS configuration errors. In 2014, Fahl et al. surveyed 755 web operators that ran websites with invalid X.509 certificates, and identified that one-third of survey participants accidentally misconfigured their webservers [13]. This misconfiguration can lead to a host of security issues, many of which the website operators were unaware of. Researchers have further analyzed misconfiguration in other contexts, such as certificate correctness [18], DNSSEC deployments [10], and enterprise networks [26].

Most similar to our work is by Krombholz et al., who measured the usability of deploying HTTPS with a laboratory study of 28 participants [17]. Our work differs primarily in that Krombholz et al. ran their study prior to the release of Let's Encrypt, which has since grown to be the most widely deployed certificate authority on the Internet [24]. In addition, our work aims to capture a broader set of participants with varying levels of technical expertise, rather than focus on those that have proficient knowledge in the areas of security and privacy. Overall, we seem to confirm the Krombholz results. Combining their results with ours, the lack of usability in HTTPS deployment seems a pervasive barrier to securing the Internet.

7 CONCLUSION

Significant work remains to be done in better understanding how users interact with many of the systems required to successfully deploy HTTPS, like terminals and text editors. Our study also would have benefited from a more precise means of expert participant recruitment, which may also prove a fertile area of future research.

Overall, we find that the technology involved in deploying HTTPS is not particularly user-friendly. Let's Encrypt did seem to better enable users to deploy HTTPS, and it also seemed speed up the process over manual configuration. However, we did not find reliable effects regarding the satisfaction of either method.

ACKNOWLEDGMENTS

The authors would like to thank Stephen Reger, Jeffrey, and our anonymous reviewers for their help and constructive feedback. This material is based on work supported by the National Science Foundation under grants CNS-1409505, CNS-1518888, CNS-1755841, CNS-1409401, and CNS-1314492.

REFERENCES

[1] certbot. https://certbot.eff.org.
[2] Namecheap. https://www.namecheap.com/.
[3] OpenSSL. https://www.openssl.org/.
[4] Surveymonkey. https://surveymonkey.com.
[5] D. Akhawe, B. Amann, M. Vallentin, and R. Sommer. Here's my cert, so trust me, maybe?: understanding tls errors on the web. In *22nd international conference on World Wide Web*, 2013.
[6] D. Akhawe and A. P. Felt. Alice in warningland: A large-scale field study of browser security warning effectiveness. In *22nd USENIX Security Symposium*, 2013.
[7] A. Bangor, P. Kortum, and J. Miller. Determining what individual sus scores mean: Adding an adjective rating scale. *Journal of usability studies*, 4(3):114–123, 2009.
[8] A. Bangor, P. T. Kortum, and J. T. Miller. An empirical evaluation of the system usability scale. *Intl. Journal of Human–Computer Interaction*, 24(6):574–594, 2008.
[9] J. Brooke. Sus: A "quick and dirty" usability scale. In P. W. Jordan, B. Thomas, B. Weerdmeester, and I. L. McClelland, editors, *Usability evaluation in industry*, page 189–194. Taylor and Francis, 1996.
[10] T. Chung, R. van Rijswijk-Deij, B. Chandrasekaran, D. Choffnes, D. Levin, B. M. Maggs, A. Mislove, and C. Wilson. A longitudinal, end-to-end view of the dnssec ecosystem. In *26th USENIX Security Symposium*, 2017.
[11] The DROWN attack. https://drownattack.com/.
[12] Z. Durumeric, D. Adrian, A. Mirian, M. Bailey, and J. A. Halderman. Tracking the FREAK attack. https://freakattack.com/.
[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 *9th ACM Symposium on Information, Computer and Communications Security*, 2014.
[14] G. Gebhart. We're halfway to encrypting the entire web. https://www.eff.org/deeplinks/2017/02/were-halfway-encrypting-entire-web.
[15] ISO. ISO 9241-11:1998 Ergonomic requirements for office work with visual display terminals (VDTs) – Part 11: Guidance on usability. Technical report, International Organization for Standardization, 1998.
[16] P. Kortum. *Usability assessment: how to measure the usability of products, services, and systems*. Human Factors and Ergonomics Society, 2016.
[17] K. Krombholz, W. Mayer, M. Schmiedecker, and E. Weippl. "I Have No Idea What I'm Doing"–On the Usability of Deploying HTTPS. In *26th USENIX Security Symposium*, 2017.
[18] D. Kumar, M. Bailey, Z. Wang, M. Hyder, J. Dickinson, G. Beck, D. Adrian, J. Mason, Z. Durumeric, and J. A. Halderman. Tracking certificate misissuance in the wild. In *39th IEEE Symposium on Security and Privacy*, 2018.
[19] B. Moller, T. Duong, and K. Kotowicz. This POODLE bites: Exploiting the SSL 3.0 fallback. https://www.openssl.org/~bodo/ssl-poodle.pdf.
[20] Netcraft. January 2018 Web Server Survey. https://news.netcraft.com/archives/2018/01/19/january-2018-web-server-survey.html.
[21] R. W. Reeder, A. P. Felt, S. Consolvo, N. Malkin, C. Thompson, and S. Egelman. An experience sampling study of user reactions to browser warnings in the field. In *2018 Conference on Human Factors in Computing Systems*, 2018.
[22] Q. SSLLabs. Ssllabs ssl server test. https://www.ssllabs.com/ssltest/.
[23] J. Sunshine, S. Egelman, H. Almuhimedi, N. Atri, and L. F. Cranor. Crying wolf: An empirical study of ssl warning effectiveness. In *18th USENIX Security Symposium*, 2009.
[24] B. VanderSloot, J. Amann, M. Bernhard, Z. Durumeric, M. Bailey, and J. A. Halderman. Towards a complete view of the certificate ecosystem. In *16th ACM Internet Measurement Conference*, 2016.
[25] S. Wells. Chrome rolls out for all users 'not secure' markers on unencrypted pages. https://techcrunch.com/2018/07/24/chrome-rolls-out-for-all-users-not-secure-markers-on-unencrypted-pages/.
[26] J. Zhang, Z. Durumeric, M. Bailey, M. Liu, and M. Karir. On the mismanagement and maliciousness of networks. In *21st Network and Distributed Systems Security Conference*, 2014.

A EXAMPLE SCRIPT FROM STUDY 1

First, can you verify that you are <insert name>?

I need you to read this Institutional Review Board consent form. <hand form and pen>. Read carefully and sign. If you have any questions let me know.

You are about to participate in a study about two different methods of deploying HTTPS. You will be acquiring a certificate and configuring a web server to serve HTTPS in two different ways. One way will be using software from Let's Encrypt, and the in the other way you will be purchasing a certificate from a certificate authority and configuring the web server by hand. After the study you will be asked some questions about your experience, and fill out a survey.

Any credentials, methods of payment, or other necessary information will be provided to you on these sheets <gesture to info sheets>, one for each method of HTTPS deployment <note differences between sheets>. The domains you will acquire a certificate for have already been purchased and registered to point to the provided server. The server software is Apache, and it has already been installed on the server. The server firewall is already disabled. An email address and password will be provided, but for any task that requires you to create a password, feel free to come up with your own. I can reset the account later if necessary. For the phase where a payment method is required, you can use the provided card as if it were a credit card, using the payment info provided. The terminal is already connected to the server via SSH.

You should not need any additional materials or information to complete the task. Imagine you're at home or work, and do whatever you would normally do to complete the task, including performing web searches, calling a friend, etc. During the study I will be unable to answer any questions, and will respond with 'do whatever you think you need to do to complete the task.'

During the study I ask that you refrain from using your cell phone to take calls. If you would like to quit at any time during the study, just say so.

If you have any questions at this point, feel free to ask. Also, if you would like to run to the bathroom, now is a good time to do so.

<Start ssh'd into server, and with blank Safari tab open>

Please tell me when you're done with the task. You may begin. <start timing on "begin">.

<After finish, or "Are you done">

<Reference questions on sheet>
<Run SSLLabs test>

You are now done with the first phase of the study. Keeping in mind the system you just used, please complete this survey, let me know when you are done..

<Swap info sheets and other materials as needed>
<Reset server (reference to the scripts)>
<Clear browser history, restart terminal and reset connection to server>
<Start ssh'd into server, and with blank Safari tab open>

If you would like to use the bathroom, now is a good time.

Now we're going to use the second system, disregard information from previous task and use this new information. Both the server and this machine have been reset, discarding any terminal and browser history.

If you are ready, please begin. <start timing on hand on mouse>.

<After finish, or "Are you done">

You are now done with the second phase. Keeping in mind the system you just used, please complete this survey.. The first part of the survey is like the one you finished after phase one, and there are more questions at the end. Let me know when you are done

<Give computer with survey>

<After survey>

You are now done! Thank you for participating in this study, your doing so has directly contributed to making the Internet a safer place. If you have any questions about the research I'm happy to answer them now (time permitting). If you think of anything after you leave, you can reach me through the information on the debrief form

<give debrief form>

And if you know anyone who would like to participate in this study, please refer them to us:

<hand flyer>

If you do refer someone, please refrain from discussing details of the study with them, as this can affect our data and impact our results.

<Fill out amount, date, have them sign, initial, then hand cash>

Thank you once again for participating!

B EXAMPLE INFORMATION SHEET

General Info
Email: blah@blah.com

Email password: <password>

Domain: usabletls1.xyz

Phone: 888-888-8888

Address: <Address of research office>

CA: namecheap.com

Payment Info

Name: <Researcher Name>

Email: blah@blah.com

Card number: provided

Phone: 888-888-8888

Address: <Address of research office>

Server Info

Server type: Apache

Server OS: Ubuntu 16.04

Server account password: <password>