"I Have No Idea What I'm Doing" –
On the Usability of Deploying HTTPS

Katharina Krombholz
SBA Research

Wilfried Mayer
SBA Research

Martin Schmiedecker
SBA Research

Edgar Weippl
SBA Research

Abstract

Protecting communication content at scale is a difficult task, and TLS is the protocol most commonly used to do so. However, it has been shown that deploying it in a truly secure fashion is challenging for a large fraction of online service operators. While *Let's Encrypt* was specifically built and launched to promote the adoption of HTTPS, this paper aims to understand the reasons for why it has been so hard to deploy TLS correctly and studies the usability of the deployment process for HTTPS. We performed a series of experiments with 28 knowledgable participants and revealed significant usability challenges that result in weak TLS configurations. Additionally, we conducted expert interviews with 7 experienced security auditors. Our results suggest that the deployment process is far too complex even for people with proficient knowledge in the field, and that server configurations should have stronger security by default. While the results from our expert interviews confirm the ecological validity of the lab study results, they additionally highlight that even educated users prefer solutions that are easy to use. An improved and less vulnerable workflow would be very beneficial to finding stronger configurations in the wild.

1 Introduction

Transport Layer Security (TLS) and its predecessor Secure Sockets Layer (SSL) are fundamental cryptographic protocols to secure information in transit across computer networks and are employed to ensure privacy and data integrity between two communicating parties. They are used hundreds of million of times every day worldwide in applications such as web browsers, smartphone apps or in email communication. Recent studies on TLS usage in the Internet ecosystem for both HTTPS [16, 25] and email [24, 31], however, revealed that a large fraction of communication endpoints are poorly secured

and suspectible to a broad array of possible attacks (e.g., Heartbleed [3] and DROWN [11]). Additionally, human-centric studies [20] have shown that warnings are still clicked through and that users have little to no understanding regarding the implications of visiting a website without a valid certificate. Even worse, a large number of services and websites still refrains from using TLS by default for all communication channels despite all efforts in propagating the use of encryption. While the initiative *Let's Encrypt* was specifically launched to offer free certificates that are trusted by all browsers, it is not yet ubiquitiously used for various reasons, e.g., the lack of Extended Validation (EV) Ceritificates. Besides that, *Let's Encrypt* still requires to be configured at least once.

To date, most studies on human-centric concerns focused on non-expert end users and, to the best of our knowledge, no user study has yet been conducted to examine the usability of the TLS deployment process directly. Our contribution aims to fill this gap by presenting the first user study with expert users to identify key usability issues in the deployment process of TLS that lead to insecure configurations. We conducted lab sessions that lasted 2 hours each with 28 university students from 14 to 18 December 2015. Data was collected via a think-aloud protocol as well as an entry and exit questionnaire. In addition we collected the bash and browser histories and the resulting server configuration files. We focused on Apache, as this is the most common web server to date [7] (A query at *censys.io* resulted in 20,890,000 websites using Apache). We found that configuring TLS on Apache is perceived as a challenging task even by experienced users. Our results suggest that administrators struggle with important security decisions (e.g., choosing the right cipher suites) which are mainly driven by concerns about compatibility. Furthermore, our participants had a hard time finding reliable sources on the Internet to

support their decision making process. The configuration options in Apache are perceived as difficult to understand and therefore an additional source of error. Through our expert interviews, we collected evidence that insufficiently secure configurations – like those from the majority of participants from our lab study – are frequently encountered during security audits. Our results shed light on major challenges from an expert user's perspective. We are confident that our results are a good baseline for the development of improved tools and policies that are better tied to the expert users' needs.

The contributions of this paper thus are:

- a **lab study with 28 participants** to explore usability challenges in the TLS configuration process

- **expert interviews with 7 security auditors** to provide a baseline for ecological validity and to further explore potential usability improvements and recommendations for the deployment process.

2 Background & Related Work

Transport Layer Security is the foundation of today's web security. Several application layer protocols use TLS to secure their online communication. The most widely used protocol is HTTPS, i.e., TLS provides confidentiality, authenticity and integrity for HTTP. Currently, TLS 1.2 [14] is the most recent version of the SSL/TLS protocol family, with TLS 1.3 on the horizon.[1] Besides securing the majority of today's web traffic, researchers have found several challenges regarding TLS, which are vigorously discussed in the literature [13,37]. Guidelines and best practices for a proper TLS deployment have also been published [12, 38]. The goals of TLS include extensibility and interoperability. This includes the ability to change the quality of the used certificate, settings of used cryptographic primitives (cipher suites), enabling of TLS extensions, use of different TLS versions and the use of additional security features like HTTP Strict Transport Security (HSTS) [23] and HTTP Public Key Pinning (HPKP) [18]. In the last years, many studies focused on empirically testing the quality of TLS configurations by using Internet-wide scanning techniques and showed that the TLS landscape is diverse and full of misconfigurations. Lee et al. [29] analyzed the supported SSL/TLS versions, the EFF started to analyze used certificates [17] with the most comprehensive study by Durumeric et al. [16] and VanderSloot et al. [42]. With a newly introduced search engine it is also possible to monitor the ecosystem more easily [15]. Ristic [36] analyzed different parameters and evaluated the quality by

a defined metric [2]. Huang et al. [26] surveyed the use of cipher suites and Kranch and Bonneau [28] scanned domains for HSTS and public key pinning.

Most user studies regarding TLS and human-computer interaction focus on non-expert end users that receive certificate warnings from their browsers. Akhawe et al. [9] performed a large-scale study on the effectiveness of SSL browser warnings and found that that these warnings have high click-through rates, i.e., 70.2% of Google Chrome's SSL warnings did not prevent users from visiting the initially requested insecure site. Harbach et al. [22] presented an empirical analysis of the influence of linguistic properties on the perceived difficulty of descriptive text in warning messages and found that the several steps can help to improve text understandability.

Several studies have been conducted to improve SSL warnings [20, 21, 41, 43]: E.g., Sunshine et al. [41] conducted a survey to examine Internet users' reactions to and understanding of current SSL warnings. Based on their findings, they designed new warnings and showed that they performed significantly better. Weber et al. [43] used a participatory design approach to improve SSL warnings. Felt et al. [21] explored reasons for higher click-through rates for SSL warnings in Google Chrome compared to Mozilla Firefox. They also showed that the design of warnings can lead users towards safer decisions.

Oltrogge et al. [33] conducted an extensive study on the applicability of pinning for non-browser software as in Android apps. They found that only a quarter of their participants understood the concept of pinning. Based on their findings, they presented a web application to support developers in making the right decisions and guiding them through the correct deployment.

Fahl et al. [19] presented the first study with system administrators and found that many of their participants wished for more simplicity, e.g., simpler interfaces and automatic certificate renewal. Their results furthermore highlight the need for a better technical education of responsible personnel. In comparison to our lab experiments, the results from Fahl et al. [19] are based on self-reported data gathered via an online questionnaire and therefore provide a baseline for our study.

3 Lab Experiments

In the following, we describe the methodology used to collect and analyze the data from the lab study.

3.1 Study Design and Procedure

In order to elicit a picture of usability challenges of TLS deployment from an administrator's point of

---

[1] `https://tools.ietf.org/html/draft-ietf-tls-tls13`

view, we conducted a series of lab experiments with 28 participants. As described in Section 3.2, we recruited students with expert knowledge in the field of security and privacy-enhancing protocols at our university who fulfilled the criteria to potentially work as an administrator or were actually working as administrators.

Our experiments proceeded as follows: After the recruitment phase, the participants were invited to the lab where they were shortly briefed about the purpose of our study. After signing a consent form, they received the study assignment as presented in Appendix A. In the given scenario, they assumed the role of an administrator of an SME who is in charge of securing the communication to an Apache web server with HTTPS in order to pass a security audit. The server system to secure was based on Raspian, a Debian-based Linux distribution. The Apache version in use was 2.4.11. We prepared and implemented a fictive Certificate Authority (CA) in order to facilitate the process of getting a valid certificate and to remove any bias introduced by the procedures from a certain CA. The fictive CA was available through a simple web interface and required the submission of a valid CSR (certificate signing request) for issuing a valid certificate. The user interface was very simplistic and the browser on the local machine already trusted our CA. Figure 2 in Appendix A shows a screenshot of the user interface. We opted for this study setting as we solely wanted to focus on the actual deployment process instead of the interaction with a CA. There was no existing TLS configuration on the system, hence the participants had to start a new configuration from scratch. We chose Apache for our experimental setup as to date, Apache maintains a clear lead regarding in usage share statistics, followed by Microsoft and nginx, e.g., [1].

We instructed the participants to make the configuration as secure as possible, whereas the assignment did not contain any specific security requirements, such as which cipher suites to use or whether to deploy HSTS or not. In order to collect data, we used a think-aloud protocol. While the participants were working on the task, they articulated their thoughts while an experimenter seated next to them observed their work and took notes. We refrained from video recording due to the results from our pre-test during which we filmed the sessions and noticed a severe impact on the participants' behavior. The participants from the pre-study also explicitly reported that they perceived the cameras as disruptive and distracting, even though they were placed in a discreet way.

In addition to the notes from the observation, we captured the bash and browser history and the final configuration files. After completing the task, the participants were asked to fill out a short questionnaire with closed- and open-ended questions which covered basic demographics, previous security experience in industry and reflections on the experiment. The complete assignment and questionnaire can be found in the Appendix of this paper.

As a result, we had a collection of both qualitative and quantitative data that was further used for analysis as described in Section 3.3.

3.2 Recruitment and Participants

In contrast to most previous studies in the area of TLS usability, we focused on users that have proficient knowledge in the field of security and privacy-enhancing technologies. As it was very difficult to recruit participants from companies, irrespective of a financial incentive, we decided to recruit participants at the university and targeted students that had previously completed a set of security courses similar to recent studies with expert users, e.g., [8, 35, 44].

To ensure that our sample reflected job requirements of real world system administrators we reviewed open job advertisements for system administrators to determine requirements for participation in our study. We then invited a selected set of students that completed several security-related courses to take an online quiz to additionally assess their knowledge irrespective of their previously issued grades. The full set of questions from the quiz can be found in Appendix A. The quiz as well as the required previously completed university courses were selected based on a review of 15 open job advertisements for system administrators in our region. The top 30 students with the best scores were then invited to participate in the lab study, and 28 of them did. The participants' completed the quiz with scores ranging from 8.21 and 10 (out of 10). The average score was 9.15 (median = 9.37). The average time to complete the quiz was 6.1 minutes.

Table 1 summarizes key characteristics of the participants: 2 participants were female, 26 were male; the age range was 21 to 32 with a median of 23. Their experience working in industry ranged from 2 to 120 months with a median of 25 months. 17 of our 28 participants were already experienced system administrators and reported to have deployed TLS before.

We are confident that our participants are suitable to explore usability challenges in TLS deployment that real-world system administrators face. To furthermore strengthen ecological validity of our results we conducted a set of interviews with security auditors (Section 5).

Demographic	Number	Percent
Gender		
Female	2	10%
Male	26	90%
Age		
Min.	21	
Max.	32	
Median	23	
Months worked in industry		
Min.	2	
Max.	120	
Median	25	
Experienced as sysadmin		
Yes	17	60%
No	11	40%
Configured TLS before		
Yes	17	60%
No	11	40%
Currently administrating		
Company web server	5	17%
Private web server	17	83%

Table 1: Participant characteristics from the lab experiments. n=28

3.3 Data Analysis

For a qualitative analysis of the observation protocols we performed a series of iterative coding which is often used in usable security research to develop models and theories from qualitative data [27, 34, 39]. Our approach involved several steps in the analysis process and was implemented as follows: At first, two researchers traversed all data segments independently point-by-point and assigned descriptive codes. This process is referred to as *open coding*. The two researchers performed the initial coding independently from each other to minimize the susceptibility of biased interpretation. We evaluated the quality of our initial codes and agreed on a final set of codes which was then used to code the protocols. Our analysis showed a good inter-rater agreement between the two coders (Cohen's κ=0.78). On the resulting initial set of coded data we performed *axial coding* to look for explanations and relationships among the codes and topics to uncover structures in the data. Then we performed *selective coding* to put the results together and derive a theory from the data.

In order to structure the data from the open-ended questions collected through the questionnaire we used an *iterative coding* process. Hence we went through the col-

lected data and produced an initial set of codes. Then we revised the retrieved codes and discussed recurring themes, patterns and interconnections. After agreeing on a final set of codes, we coded the entire data. As a result of our analysis, we obtained a picture of usability challenges in the deployment process which is presented in Section 4, grouped by themes.

To evaluate the (mostly) quantitative data acquired via the bash/browser history and Apache log files, we applied metrics and measures to evaluate the quality of the resulting configuration.

4 Results

In this section we present the results from our lab study which are based on the data from the think-aloud protocol, the collected log files and the self-reported data from the exit-questionnaire.

4.1 Security Evaluation

We based our evaluation criteria on Qualy's SSL Test.[2] We consider this rating scheme a useful benchmark to assess the quality of a TLS configuration based on the state of the art recommendations from various RFCs [37, 38] and with respect to the most recently discovered vulnerabilities and attacks in the protocol. Since web services have different requirements, e.g., backward compatibility for outdated browsers, there is no universally applicable recommendation to get the highest grade. Still, the rating is widely accepted and applicable to generic web services like in our study. It must be mentioned that this benchmark reflects the best-case scenario at the time of writing, but could be different in the future if new vulnerabilities are discovered.

The rating of the evaluation criteria is expressed with a grade from A to F and composed out of three independent values: (1) protocol support (30%), (2) key exchange (30%) and (3) cipher strength (40%). Some properties, e.g., support for the RC4 cipher cap the overall grade as shown in Table 3. Table 2 summarizes the results of a security evaluation based on the final configuration per participant with additional information in Table 3. The full set of evaluation criteria based on the metrics used in Qualy's SSL Test is listed in Appendix A.

Only four participants managed to deploy an A grade TLS configuration, P24 received the best overall score. B was the most commonly awarded grade (15 out of 28). Four participants did not manage to deploy a valid TLS configuration in the given time (P7, P18, P23, P26). Two

---

[2] https://www.ssllabs.com

participants (P10 and P19) encrypted their private keys, the passphrases were "abc123" and "pass". One of these two did not share the passphrase with us, however it was easy to brute-force.

Fortunately, none of our participants chose a key size smaller than 2048 for their RSA key. 15 participants chose 2k- and eight chose 4k-sized keys. Five out of the 28 participants deployed the certificate chain correctly, which is necessary to receive a grade better than B according to our rating scheme.

Two participants did not make use of the study CA and used self-signed certificates. Only one participant enabled a TLS version lower than TLS 1.0 (P8), another participant had all versions but TLS 1.2 disabled (P14). Only two participants configured RC4 support and only one configuration (P8) was vulnerable to the POODLE attack as SSL 3 was still supported. 14 participants fully configured forward secrecy, the remaining participants with valid configurations managed to at least partially support it. Eleven participants included HSTS headers to improve the security of their configuration and only two participants deployed HPKP.

To determine whether the distribution of SSL Test grades from our lab study reflects those from configurations found in the wild, we consider the estimation from *SSL Pulse* [6] who regularly publishes data sets of grade distribution measures based on the Alexa Top 1 Million. This data set as of the time our study was conducted contains 141.890 surveyed sites of which 34.1% were graded with A, 20.2% with B, 27.1% with C and 18.5% failed. Based on the 24 valid configurations from our study, 25% of the study configurations were graded with A, 67% with B and 8% with C. Given that the data set from SSL Pulse [6] contains websites with potentially higher security requirements or sites were administrators were presumably given more time to obtain a secure configuration. In particular the possibly very complex structures of real-world websites, as well as the inclusion of third-party content, make our study non-representative.

4.2 TLS Deployment Model

Our qualitative analysis of the think-aloud protocols from our lab study yielded a process model for a successful TLS configuration. All participants who managed a valid configuration in the given time can be mapped to the stages presented in this model. The four participants who did not manage to deploy TLS in the given time significantly deviate from this model. We divide the steps from our model into two phases, a *setup phase* and a *hardening phase*. We refer to the *setup phase* as to a set of tasks to get a basic TLS configuration, i.e., the service is reachable via https if requested. The *hardening phase* comprises all necessary tasks to get a con-
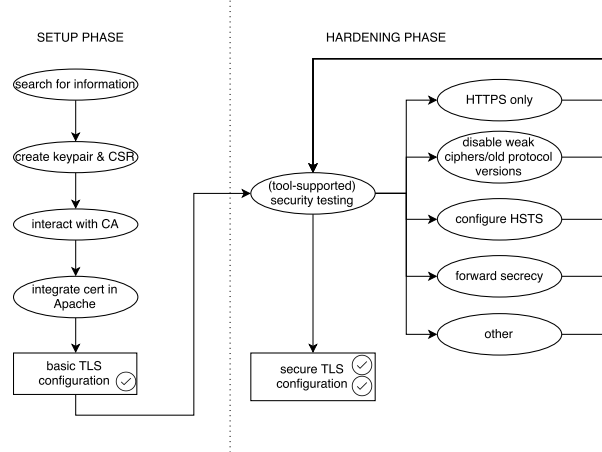

Figure 1: Schematic representation of a successful workflow.

figuration which is widely considered *secure* with respect to the metrics defined in A. Figure 1 shows our deployment model. Participants who achieved at least a basic configuration successfully completed all steps of the setup phase, while better-graded configurations completed some steps from the hardening phase as well. We identified iterative (tool-supported) security testing as a key element for a successful hardening phase, since the participants relied on external sources to evaluate the quality of their configuration.

4.3 Usability Challenges in TLS Deployment

In the following, we present the usability challenges identified through our analysis of qualitative data from the think-aloud protocols and the quantitative data from the collected log files.

Searching for information and finding the right workflow. Except for 3 experienced participants, who explicitly searched for tutorials they were aware of (e.g., `bettercrypto.org`), the study participants visited a high number of websites and used multiple sources of information. The information sources were diverse regarding their suggested deployment approaches and information quality respectively. We frequently observed that a participant started to follow an approach from one tutorial and soon had to switch to another as the presented approach was not feasible for our deployment scenario and the given server configuration.

The lowest number of visited websites during the lab study was 20 (P21). In contrary, participant P4 visited 147 websites during the given time. The average

Table 2: Security evaluation of the final TLS configuration per participant.

ID	Grade	Errors / Warnings / Highlights	Cipher Strength Score	Key Exchange Score	Protocol Support Score	Common Name	Key Size	Certificate Chain Length	Used Provided CA to Sign	Encrypted Private Key	SSL 2	SSL 3	TLS 1.0	TLS 1.1	TLS 1.2	RC4 Support	Vulnerable to POODLE (SSL 3)	Forward Secrecy	HSTS	HPKP
P1	A	2	90	90	95	web.local	4096	3	●	○	○	○	●	●	●	○	○	●	●	○
P2	B	3	90	90	95	web.local	2048	1	●	○	○	○	●	●	●	○	○	●	○	○
P3	B	2,3	90	90	95	web.local	2048	1	●	○	○	○	●	●	●	○	○	●	●	○
P4	A		90	90	95	web.local	2048	3	●	○	○	○	●	●	●	○	○	●	○	○
P5	B		90	90	95	web.local	4096	1	●	○	○	○	●	●	●	○	○	●	●	○
P6	B	3	90	90	95	web.local	2048	1	●	○	○	○	●	●	●	○	○	●	○	○
P7	Not valid																			
P8	C	3-6,8	90	90	50	web.local	2048	1	●	○	○	●	●	○	○	●	●	◐	○	○
P9	B	1-3	100	90	95	web.local	4096	1	●	○	○	○	●	●	●	○	○	●	●	●
P10	B	1-3	90	90	95	web.local	4096	1	●	○	○	○	●	●	●	○	○	●	●	●
P11	B	3,4	90	90	95	web.local	2048	1	●	●	○	○	●	●	●	○	○	◐	○	○
P12	B	2,3	90	90	95	web.local	4096	1	●	○	○	○	●	○	●	○	○	●	●	○
P13	B	3	90	90	95	web.local	2048	1	●	○	○	○	●	●	●	○	○	◐	○	○
P14	A-	4	90	90	100	raspberrypi	2048	1	○	○	○	○	○	○	●	○	○	◐	○	○
P15	C	4,7	50	90	95	-	2048	1	○	○	○	○	●	●	●	●	○	◐	○	○
P16	A-	4	90	90	95	web.local	2048	3	●	○	○	○	●	●	●	○	○	◐	○	○
P17	B	2,3	90	90	95	web.local	3096	1	●	○	○	○	●	●	●	○	○	●	●	○
P18	Not valid																			
P19	B	2,3	90	90	95	web.local	2048	1	●	●	○	○	●	●	●	○	○	●	●	○
P20	B	2,3	90	90	95	web.local	2048	1	●	○	○	○	●	●	●	○	○	●	●	○
P21	B	3,4	90	90	95	Test	2048	1	●	○	○	○	●	●	●	○	○	◐	○	○
P22	B	3,4	90	90	95	web.local	2048	1	●	○	○	○	●	●	●	○	○	◐	○	○
P23	Not valid																			
P24	A	2	90	90	97	web.local	2048	3	●	○	○	○	○	●	●	○	○	●	●	○
P25	B	3	90	90	95	SME	4096	1	●	○	○	○	●	●	●	○	○	◐	○	○
P26	Not valid																			
P27	B	3,4	90	90	95	web.local	4096	1	●	○	○	○	●	●	●	○	○	◐	○	○
P28	A	2	90	90	95	web.local	4096	3	●	○	○	○	●	●	●	○	○	●	●	○

1	Highlight	HTTP Public Key Pinning (HPKP) deployed on this server. Yay!
2	Highlight	HTTP Strict Transport Security (HSTS) with long duration deployed on this server.
3	Warning	This server's certificate chain is incomplete. Grade capped to B.
4	Warning	The server does not support Forward Secrecy with the reference browsers.
5	Warning	This server accepts RC4 cipher, but only with older protocol versions. Grade capped to B.
6	Warning	The server supports only older protocols, but not the current best TLS 1.2. Grade capped to C.
7	Warning	This server uses RC4 with modern protocols. Grade capped to C.
8	Error	This server is vulnerable to the POODLE attack. If possible, disable SSL 3 to mitigate. Grade capped to C.

Table 3: Errors / Highlights / Warnings as referred to in Table 2.

number of visited websites over all participants was 60 (median=49.5, sd=27). We consider this a relatively high number given the low amount of time. Table 5 lists the most visited websites. The top-most visited site points to a German Ubuntu and Linux wiki that is frequently updated. The documentation for SSL on Apache (second-most visited site) contains detailed information on certificate creation and retrieval but only basic information on hardening. In contrast, `sslabs.com` and `bettercrypto.org` contain comprehensive tutorials on hardening but require a detailed understanding of the underlying fundamentals. The tutorial from `raymii.org` provides step by step instructions but is not regularly updated. Most participants expressed annoyance and vexation about the incompatibility of the different information sources. We also found that the number of visited websites (high, medium, low) does not impact the quality of the resulting configuration, but this result is not significant in our sample with $\chi^2(0.23327892, 6) > 0.05$.

"I have absolutely no idea what I'm doing. Neither am I aware of whether my online source is trustworthy. (P23)"

Creating a Certificate Signing Request (CSR). A CSR is a block of PEM-encoded text which is sent to a CA to request a TLS certificate. It therefore contains information that will be included in the certificate such as organization name and common name (FQDN) and enables users to send their public key along with some information that identifies the domain name in a standardized way. When creating a CSR, the user is asked to fill out the respective information. In order to create a CSR, the user has to create a key pair. Our results suggest that many users do not understand the purpose and concept of a CSR, i.e., who it is authenticating towards whom. 19 out of 30 participants from the lab study had to create two or more requests due to errors in the CSR creation. The most common error was that they did not fill out the requested common name field correctly (14 participants) and thus did not receive a valid certificate for their domain. In the end, 20 participants created a CSR

Participant ID	Visited websites	Grade
Most visited sites		
P4	147	A
P19	116	B
P8	111	C
P2	109	B
P7	116	-
Least visited sites		
P21	20	B
P12	36	B
P5	49	B
P10	49	B
P18	50	-

Table 4: Participants and their cumulative number of visited sites and overall rating.

with the correct common name as shown in Table 2. As this is a common error in practice, some CAs even highlight that the common name(s) can be altered later on. This is especially useful when adding TLS support for subdomains. Second, two participants (P14 and P15) did not fully understand the difference between a CSR and a (self-signed) certificate. Six participants initially created a self-signed certificate instead of a CSR and tried to upload it to the CA. According to the self-reported work experience, this happened to participants regardless of their experience. E.g., P15 reported to have recently deployed TLS on Apache and still tried to upload a self-signed certificate to the CA. Four participants recognized the error after receiving an error message from the CA and then created a correct CSR including a correct common name.

Choosing the appropriate cipher suites. In TLS, cipher suites are used to determine how secure communication takes place. Cipher suites are composed from building blocks in order to achieve security through diversity. A person in charge of configuring TLS has to select cipher suites that provide authentication and en-

URL	Visitors
`wiki.ubuntuusers.de/Apache/SSL`	25
`httpd.apache.org/docs/2.4/ssl/`	20
`www.ssllabs.com/`	16
`bettercrypto.org`	15
`raymii.org/s/tutorials/Strong_SSL_Security_..`	14
`httpd.apache.org/docs/2.2/mod/mod_ssl`	11

Table 5: Top most visited websites.

cryption that is considered strong. However, this is a task that requires a deep and up-to-date understanding on the underlying algorithms in order to make informed decisions about which cipher suites to support. In the course of our lab experiments, all participants who came to this point during the configuration assignment were aware of the fact that they had to manually select cipher suites to secure the communication. The decision making process was exclusively based on search results and suggestions from online resources without questioning. Some participants also referred to recently published blog posts where they read about the disadvantages of a certain algorithm. This implies that the quality of the used information source is crucial for the overall security of the configuration as our participants lacked profound knowledge and thus had to trust their source of information. Table 2 shows how the selected cipher suites impact the quality of the configuration.

Strict HTTPS. After finishing an initial valid configuration, most participants enforced strict HTTPS as a first step of the hardening phase. Some were annoyed by the fact that HTTPS does not immediately replace HTTP as soon as it is available. Most participants were initially confused when they tested their configuration via the browser and were redirected via http when they entered the URL without the http(s):// prefix. They then spent a significant amount of time to configure the virtual host and the respective ports correctly, mostly also due to misleading or incomplete information from online sources.

Multiple configuration files. All but six participants said that they found the configuration file structure confusing, regardless of their prior experience with Apache. P14 found it particularly challenging to find the right configuration files. According to the think-aloud protocol, this was the main challenge that in the end resulted in an invalid configuration. Several participants copied and pasted entries between different configuration files or had double entries, e.g., for *SSLEngineOn*. Nine participants also struggled with loading the modules, e.g.,

P18 did not understand where to load the modules in the configuration. Many participants were also not aware of where and how to create a new virtual host which listens on 443. P23, for example, did not understand the differences between the http.conf and apache.conf which distracted him/her from the TLS-specific tasks and security-critical decisions.

Finding the right balance between security and compatibility. We observed that the majority of our participants struggled with the definition of a *secure* configuration. In our assignment we just stated that the configuration should be *as secure as possible to withstand an audit*, without specifying any key properties. Hence, the participants themselves had to make the decisions. About 15 participants expressed concerns regarding compatibility when configuring SSL/TLS versions and cipher suites. A majority of them, however, decided in favor of a securer option, e.g., disabling all TLS versions < TLS v1.1 and thus refraining from supporting older versions of IE.

4.4 Impact of Prior Experience with TLS

As shown in Table 1 a significant proportion of the participant pool has already administered or is currently administering a server and 17 participants have configured TLS before. Regardless of our relatively small sample which is due to the qualitative nature of our study, we provide statistical significance of the interplay between prior experience and the resulting security grade from our study. Table 6 shows the cumulative amount of participants that achieved a certain security grade during the study with respect to their prior experience. None of the participants who did not manage to provide a valid configuration in the given time had prior experience with server administration in a corporate environment. However, Table 6 shows that the majority of experienced users was not able to provide an A grade configuration. A significance test with $\chi^2(7.9982, 3) = 0.046$ provides evidence to suggest that there is an association between prior experience with configuring TLS and the

grade of a participant's TLS configuration from the lab study. We could not identify dependence between prior employment as system administrator and the SSL Test grade based on the configuration from the lab study with $\chi^2(6.7667, 3) = 0.07$.

4.5 Perceptions of Usability

After the lab experiments, the study participants filled out a short online questionnaire and reported reflections on the assignment. 18 participants reported that they thought they finished the assignment completely, while nine thought that there were still some configuration steps missing. One participant was not sure about whether or not he/she finished the task. While ten participants perceived the assigned task as difficult and three as very difficult, only four participants thought that it was easy and one that it was very easy. Twelve participants rated the difficulty as neutral.

We also asked our participants what they think are the most severe usability pitfalls in the deployment process. In the following, we provide a respective list. Most frequently mentioned were lack of best practice tutorials (19), followed by misleading terminology (15) and weak default configurations (12).

Lack of best practice tutorials. According to our participants, it was difficult to determine a best practice on how to deploy TLS. Our participants reported that they came across outdated or simply wrong information in online tutorials. 13 participants also mentioned that most tutorials were not generic, but still not specific enough to apply them to the system given in the assignment.

Misleading terminology and error messages. Especially with respect to interactions with the CA, participants expressed confusion about the terminology. Some accidentally uploaded a self-signed certificate instead of a CSR and found the file endings difficult to handle and to distinguish, e.g., *.key*, *.pem*, *.crt*.

Weak default configuration. Eight participants explicitly criticized the high effort necessary to harden the configuration, as too many cipher suites are enabled by default. Also, they criticized that the selection of cipher suites is a time-consuming task that requires profound background knowledge in order to make an informed decision and that bad decisions yield major security vulnerabilities. One participant also suggested a simplified configuration option including a two- or three-way variable to disable certain cipher suites (e.g., tinfoil hat vs. maximum compatibility). Four participants also

stated that they would prefer if web servers had TLS configured by default.

"It seems that there is already a certificate called snakeoil, why can't I use this one?" (P7)

Confusing config file structure. During the configuration process, many participants perceived the Apache config file structure as confusing and experienced it as a severe source for errors. We also observed that some participants had simple copy/paste errors in their config files which highly distracted them from the actual main task.

"There are multiple config files in /etc/apache2, how and where do I have to load modules?" (P18)

"Why is there a snakeoil certificate in the config file?" (P22)

Complex workflow. Six participants explicitly stated that the workflow itself is too complex due to the different approaches and branches that can be taken during the configuration process as well as the dependencies of the subtasks. Three participants stated these factors hindered them in finding the source of an error afterwards.

"The configuration process is fiddly and one has to google tons of pages to get it right. Even then one cannot be sure to have a good configuration because SSL vulnerabilities are discovered almost on a regular basis." (P9)

Too much background knowledge required. Many participants expressed their concern about the high amount of background knowledge required to successfully configure TLS in a secure way. Also, the fact that a TLS configuration must be well maintained and frequently updated requires the person in charge to be informed about the latest TLS attacks and other vulnerabilities which our participants considered infeasible in practice.

Confusing permissions. Five participants also stated that they found it hard to choose the correct location and permissions for the certificate and private key.

5 Expert Interviews

In order to address ecological validity, we conducted additional expert interviews with security consultants and auditors about their experiences with insecure TLS configurations. In this section, we describe the interview

Experience	A	B	C	not valid
Configured TLS before?	5	11	1	0
Worked as admin in the past	4	4	0	0
Administering company server	1	3	1	0
Private server	4	9	0	2

Table 6: Prior experience with TLS deployment and server administration.

methodology and results of these expert interviews that were conducted in April 2016. The interview guideline can be found in Appendix A.

5.1 Recruitment and Interview Procedure

The participants were recruited at a security conference in Germany with participants from both academia and industry and via emails to regional security consulting companies. The requirements for participation currently work as a security consultant or auditor and to have at least 2 years of prior experience in auditing web services. The expert interviews were conducted as semi-structured interviews with 7 security experts from well-respected security consulting firms in the German-speaking region. The experts were familiar with TLS misconfigurations and frequently encountered misconceptions on how to combat the trade-off between compatibility and security. The interview segments were coded using iterative coding.

5.2 Results

Our results show that auditors commonly agree that poor usability and too complex workflows and server configurations result in weak TLS configurations. They also mentioned that the deployment process must be simplified and especially the default configuration should favor security. In the following, we discuss their responses in detail. Six interview participants were male, one was female. The average number of months spent as a penetration tester or auditor was 53.2. Two participants work in small companies with less than 10 employees, the remaining participants were employed in companies with more than 10 but less than 100 employees.

Auditing TLS configurations. All expert interview participants reported to focus on the following configuration characteristics during audits: activated TLS/SSL version, activated cipher suites, if the certificate is recognized by commonly used web browsers, whether HSTS is configured and whether public key pinning is activated. E3 and E7 also highlighted that they particularly pay attention if recently discovered attacks are mitigated. E6

and E7 also said that in addition to automated tools, they prefer to evaluate the server configuration directly, if it is accessible.

All seven interview participants use Qualy's SSL Test as the de-facto standard to evaluate public domains. They also use selected Nessus modules[3] and OpenVAS[4] for internal sites. E2, E4 and E6 also reported to use NMap [30].

Configuration mistakes in the wild. According to the interview participants, the main concern when deploying TLS is compatibility. Our interviewees, however, also mentioned that in most cases the compatibility challenge is just a mock argument which is often used as an excuse and not fully elaborated by the responsible employees. Compatibility is a challenge for publicly available sites where almost any client may want to access. However, it is a rather easy-to-solve problem for services that are only accessed internally, hence the set of potentially accessing clients is well known. Also, backward compatibility with older client versions (i.e., <IE7) may not be desired for a variety of reasons beyond TLS and will only affect a minority of clients. However, E1 and E3 also reported that finding the best fit between security and compatibility is hard even for security experts and often arguable. Five of the interviewed auditors also reported that they often find self-signed certificates which do not fulfil the intended purpose. E1, E2, E3 and E7 mentioned that they often encounter weak default TLS configurations with poor ciphers and no additional security measures (e.g., HSTS).

Two auditors mentioned that in the course of looking at TLS configurations for many years, they have never encountered HTTP public key pinning during an audit. Also, one interview participant reported that TLS deployment is not sufficiently streamlined in companies. According to them, most companies have multiple servers with varying configurations and each one is maintained and updated separately.

E2 also highlights that the ideal TLS configuration has changed frequently in the last two years or algo-

---

[3] https://www.tenable.com/products/nessus-vulnerability-scanner
[4] http://www.openvas.org/

rithms have been deprecated which implies a significant overhead for administrators to keep their configurations up to date. E2, E4 and E7 also reported that companies do not fully make use of the online sources available, such as using Qualy's SSL test for public domains.

"In most cases backward compatibility is the show-stopper regarding proper TLS configurations." (E3)

Concerns in the wild. We also asked our interviewees about the concerns that admins, CSOs and other persons in charge have regarding TLS. Our experts agreed that especially administrators are aware that configuring TLS is a sensitive task during which several things can go wrong. However, lack of time seems to be a major issue and administrators often do not have the resources to get a deep understanding on the fundamentals. To our surprise, E4 and E7 reported that they frequently encounter responsible persons that have little or no experience with security protocols such as TLS. All interview participants reported that in the course of security audits, they also frequently find weak default configurations along with little awareness regarding the weakness of such configurations and how they could easily be hardened. E7 highlighted that responsible persons even report that they are "afraid of using crypto". As an example (described in 5.2), E1 explicitly mentioned HSTS which is easy to deploy and has no impact on compatibility, but is rarely used in practice.

Also, compatibility still remains a key concern as lack of compatibility often leads to overloaded help lines, as reported by E1, E6 and E7. Also, the risk of MITM attacks is often underestimated and companies do not perceive themselves as targets of such attacks. E7 cited an administrator from an SME saying: "Our configuration supports basic encryption, so this should be more than enough... and clearly is better than no encryption." As E2 reports, companies are often concerned about introducing encryption due to the additional performance overhead which is in their opinion not worth the effort.

Suggested usability improvements. A common opinion of all interviewees was that the default server configurations must be improved by simplifications and default security options. They said that server configurations should be secure by default, i.e., that TLS should be enabled by default and hence must be explicitly disabled if necessary. E1 highlighted that Apache has a weak default configuration for compatibility reasons and mentioned the Caddy web server[5] as a good and usable example. Caddy comes with a TLS configuration by default and

uses *Let's Encrypt* to get certificates. Also, according to E1 the by default activated cipher suites are a good compromise, and even OCSP stapling and HSTS are deployed by default. Also, the Caddy web server automatically renews certificates. E1 highlights that configuration directives must be simplified to yield strong configurations and that Caddy web server is a good example for this paradigm. E1 also suggests that compatibility flags which administrators can use to configure cipher suites would be much more helpful than letting them deal with cipher suites directly.

Regarding the deployment process in larger enterprises that maintain multiple servers, E1 proposes to create a strong sample configuration on a test server and to then deploy them on all servers. This potentially helps to avoid outdated configurations, as the updating process is simplified and the person in charge is aware of the TLS configuration on all devices by knowing the essentials of the sample configuration.

E1 also suggests to deploy everything that does not result in lower compatibility, i.e., OCSP stapling which is commonly ignored by clients who do not understand the according header. While public key pinning is rather difficult to fully deploy, it can easily be used in report-only mode and thus enables to detect MITM attacks. E1 highlights that these additional functionalities are beneficial for security but rarely encountered in the wild.

E3 also suggests that HTTPS should fully replace HTTP to solve security problems. E3 also thinks that HTTP has no fundamental benefit over HTTPS with TLS. E3 shifts the responsibility from servers to clients and stated that clients should be frequently updated to support the respective ciphers. Furthermore, E3 argues that the concept behind CAs also has its flaws, i.e., lack of certificate transparency, certificate revocation and lawful interception on the CA's side without the end user's consent. She/he also claims that browsers generally trust a high number of CAs with varying trustworthiness.

E7 highlighted the need for professional education and that "doing it right" requires experienced professionals that keep track of the ongoing changes. E7 also suggested that there is a high demand for better configuration guides and easier-to-use default configurations to compensate the lack of know-how of the persons in charge as well as to make it easier for everyone to configure TLS in a secure yet compatible way. Also, this interview participant said that companies should have policies regarding encryption and compatibility to make it easier for administrators to choose the right configuration.

6 Discussion

While related work already showed that TLS configurations in the wild are often weak and thus do not suf-

---

[5]`https://caddyserver.com/`

ficiently protect Internet users from MITM attacks, our work explores the reasons for this. In comparison to most related user studies, we focus on the expert user role instead of the non-expert end user who is mostly unaware of potential risks and clicks through warnings which are often hard to understand and do not sufficiently communicate security risks.

We were surprised by the helplessness that we encountered during the lab study. The security auditors who participated in our expert interviews draw a similar picture of the administrators' reaction when confronted with the results of an audit which strengthens the ecological validity of our results.

For our sample, we selected top students that successfully completed security courses and proved their technical knowledge in an initial knowledge survey. 17 out of 28 participants were already experienced with managing servers in a corporate environment. We also compare the technical knowledge of our participants with those from Fahl et al. [19] who surveyed 755 webmasters. Their results suggest that webmasters often lack of a detailed understanding of the SSL security features and that they are not sufficiently educated. Fahl et al. [19] also found that real world webmasters heavily rely on online sources to compensate for their lack of knowledge.

Based on this comparison and the results from our expert interviews we are confident that our sample is suited to explore usability challenges and reflects the diverse knowledge of administrators in the wild.

Our results suggest that poor usability is a key issue and by far the main reason for weak configurations. Through both our lab study and the expert interviews we found that even professionals lack the knowledge regarding the underlying cryptographic fundamentals such as cipher suites and even basic concepts like the role of certificates. This result shows that there is a high demand for better default configurations and/or tool support to prevent administrators from dealing with mechanisms they cannot fully understand.

As stated in Section 4.1, we based our evaluation criteria on Qualy's SSL Test to evaluate the configurations from our lab study. Although these metrics are considered a good benchmark to assess TLS configuration, not all of them are feasible for every real-world scenario. For example, HPKP in theory is a mechanism to mitigate MITM attacks with fraudulent certificates but poses additional risks and challenges in practice as key management for HPKP is hard to manage for long tail websites. Possible solutions are to pin the CA certificate and to use a backup key or to use CAA (Certification Authority Authorization) DNS records to allow domain owners to specify which CAs are allowed to issue certificates for the respective domain. During our lab experiments, two participants started deploying HPKP. However, from the data we collected during the experiments, it is unlcear to what extent the participants who wanted to deploy HPKP were aware of the implied key management challenges.

In December 2015, the initiative *Let's Encrypt* released its non-profit CA that provides free domain-validated X.509 certificates and software to enable installation and maintenance of these certificates was launched to make it easier for administrators to deploy TLS. Since then, *Let's Encrypt* changed the TLS market signifcantly. It issued over 27 million active certificates for over 12 million registered domains (Feb. 2017). It is often called the largest CA, but is still not clear how much this influenced the TLS ecosystem, since many certificates are used for less popular web sites [4, 5]. However, *Let's Encrypt* is not directly improving TLS configurations. It seems that the prime goal, the process of certificate issuance was improved, but the full TLS configuration is still a manual process. Some plugins (e.g., for Apache integration) automatically set some TLS configuration parameters (e.g., protocol version, cipher suites) to a balanced configuration in terms of security and backward compatibility. However, it does not include other parameters like HSTS or the DH prime configuration. Therefore, configurations with certificates issued by Let's Encrypt are not generically comparable with other configurations, but it is clearly an opportunity to also improve and automate the configuration process in the future. Hence, *Let's Encrypt* does not entirely automate the workflow as presented in Figure 1. In fact it aimes to ease the creation of a CSR and the interaction with the CA. Regardless of these substantial improvements, *Let's Encrypt* needs to be configured at least once. While there are dedicated tools available (e.g. ACME) it remains to show to what extent the initial effort in configuring an Apache web server actually decreases.

As mentioned by our security experts, there are already servers with a focus on better security: they let their users make configurations less secure if desired instead of providing no security by default and thus forcing users to deploy security themselves. Also, they highlight the demand for easier user interfaces for configuration purposes which corresponds to the findings of Fahl et al. [19]. Our results also suggest that expert users are often unable to decide on the appropriate level of security, which highlights the need for cross-organizational guidelines and policies.

As creating a basic TLS configuration also involves complex decisions (such as choosing the appropriate key length) it is difficult for administrators to maintain or correct errors and wrong decisions.

Both the results from the lab study and the expert interviews highlight that the complex deployment process should be simplified, and that the difference between a basic correct configuration and a secure one should not

be too broad. Hence we suggest that newly designed servers and/or supportive tools should merge the setup and the hardening phase resulting in a best-case working configuration if all steps are completed – which can then be downgraded if necessary.

6.1 Limitations

A severe limitation of our lab study is that we only looked at the initial deployment process and excluded long-term maintenance effects, such as certificate renewal and the administrators' reactions to newly discovered vulnerabilities. The main reason is that it is difficult to reliably study long-term effects in the lab. In the future, we plan to conduct an additional case study in a corporate environment to observe long-term effects over a number of years. Also, as our study was performed in the lab, the participants did not have a deep background of the notional company they were administrating for the study. Our primary goal was to recruit participants who were fully employed as system administrators, but unfortunately did not manage to get enough responses respectively commitments for participation. Therefore, we chose to recruit participants among our computer science students. To overcome this bias, we selected top students that successfully completed security courses with good grades and completed an initial assessment test. As our results suggest, many of them were already experienced with managing servers and some had even worked as system administrators in companies and other organizations. We therefore believe that our data is suited to explore usability challenges. Our expert interviews with security auditors underline the ecological validity of the results from our lab study and suggest that configurations found in the wild are even less secure than those generated by our participants during the lab study. Another limitation of our study is that we instructed the participants to deploy the securest possible configuration. This goal could be unrealistic in a corporate environment where compatibility is a major concern. Therefore our results represent an upper bound for security.

7 Ethical Considerations

Our university located in central Europe unfortunately does not have an ethics board but has a set of guidelines that we followed in our research. Also, we aligned the methodology for our user study in related studies with similar ethical challenges [35, 40, 44].

A fundamental requirement of our university's ethics guidelines is to preserve the participants' privacy and to limit the collection of person-related data as far as possible. Therefore, every study participant was assigned an ID which was used throughout the experiment and for the online questionnaire. All participants signed consent forms prior to participating in our study. The consent form explained the goal of our research, what we expected from them and how the collected data was used. The signed consent forms were stored separately and did not contain the subsequently assigned IDs to make them unlinkable to their real names.

We refrained from video-recording the participants during the study as the participants from our pre-study reported that the awareness of being filmed made them feel uncomfortable and had a negative impact on their performance even if the camera was positioned in a non-obtrusive way.

8 Conclusion

We conducted a lab study with 28 participants to explore usability challenges in the TLS deployment process that lead to insecure configurations. In comparison to related work, we contributed a study that focuses on expert users, i.e., administrators who are in charge of securing servers. Additionally, we conducted seven expert interviews with penetration testers and security auditors who frequently encounter poorly secured servers during security audits.

We found that the TLS deployment process consists of multiple critical steps which, if not done correctly, lead to insecure communications and put Internet users at risk for MITM attacks. Furthermore, our results suggest that even computer scientists who are educated in terms of privacy-enhancing protocols and information security need additional support to make informed security decisions and lack an in-depth understanding of the underlying cryptographic fundamentals. Expert users also struggle with the configuration file structure of Apache web servers and have to put a lot of additional effort into securing default configurations. Our expert interviews underline the ecological validity of the results from our lab study and shed light on the weaknesses of TLS configurations found in the wild. According to our security auditors, the main concern regarding TLS is interoperability. They also highlighted that server infrastructures are often configured with poor defaults and badly maintained and are therefore not up-to-date.

References

[1] 2016 Web Server Survey. Online at `https://news.netcraft.com/archives/2016/02/22/february-2016-web-server-survey.html`.

[2] SSL Labs Server Rating Guide. Online at `https://www.ssllabs.com/downloads/SSL_Server_Rating_Guide.pdf`.

[3] The Heartbleed Bug. Online at `https://heartbleed.com`, 2014.

[4] Is Let's Encrypt the Largest Certificate Authority on the Web? Online at `https://www.eff.org/deeplinks/2016/10/lets-encrypt-largest-certificate-authority-web`, 2016.

[5] Let's Encrypt Stats. Online at `https://letsencrypt.org/stats/`, 2016.

[6] Survey of the SSL Implementation of the Most Popular Web Sites. Online at `https://www.trustworthyinternet.org/ssl-pulse/`, 2016.

[7] Usage statistics and market share of Apache for websites. Online at `https://w3techs.com/technologies/details/ws-apache/all/all`, 2016.

[8] Y. Acar, M. Backes, S. Fahl, D. Kim, M. L. Mazurek, and C. Stransky. You Get Where You're Looking for: The Impact of Information Sources on Code Security. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 289–305, May 2016.

[9] D. Akhawe and A. P. Felt. Alice in Warningland: A Large-Scale Field Study of Browser Security Warning Effectiveness. In *USENIX Security Symposium*, pages 257–272. USENIX Association, 2013.

[10] N. J. AlFardan, D. J. Bernstein, K. G. Paterson, B. Poettering, and J. C. Schuldt. On the Security of RC4 in TLS. In *USENIX Security Symposium*. USENIX Association, 2013.

[11] N. Aviram, S. Schinzel, J. Somorovsky, N. Heninger, M. Dankel, J. Steube, L. Valenta, D. Adrian, J. A. Halderman, V. Dukhovni, et al. DROWN: Breaking TLS using SSLv2. In *USENIX Security Symposium*. USENIX Association, 2016.

[12] W. Breyha, D. Durvaux, T. Dussa, L. A. Kaplan, F. Mendel, C. Mock, M. Koschuch, A. Kriegisch, U. Pöschl, R. Sabet, B. San, R. Schlatterbeck, T. Schreck, W. Alexander, A. Zauner, and P. Zawodsky. Applied Crypto Hardening. Online at `https://bettercrypto.org`, 2015.

[13] J. Clark and P. C. van Oorschot. SoK: SSL and HTTPS: Revisiting past challenges and evaluating certificate trust model enhancements. In *2013 IEEE Symposium on Security and Privacy (SP)*, pages 511–525. IEEE, 2013.

[14] T. Dierks and E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246 (Proposed Standard), Aug. 2008. Updated by RFCs 5746, 5878, 6176.

[15] Z. Durumeric, D. Adrian, A. Mirian, M. Bailey, and J. A. Halderman. A search engine backed by Internet-wide scanning. In *Conference on Computer and Communications Security*, pages 542–553. ACM, 2015.

[16] Z. Durumeric, J. Kasten, M. Bailey, and J. A. Halderman. Analysis of the HTTPS Certificate Ecosystem. In *Internet Measurement Conference*, pages 291–304. ACM, Oct. 2013.

[17] P. Eckersley and J. Burns. An Observatory for the SSLiverse. DEF CON 18 `https://www.eff.org/files/defconssliverse.pdf`, July 2010.

[18] C. Evans, C. Palmer, and R. Sleevi. Public key pinning extension for HTTP (HPKP). RFC 7469, 2015.

[19] S. Fahl, Y. Acar, H. Perl, and M. Smith. Why Eve and Mallory (Also) Love Webmasters: A Study on the Root Causes of SSL Misconfigurations. In *Proceedings of the 9th ACM symposium on Information, Computer and Communications Security*, ASIA CCS '14, pages 507–512, New York, NY, USA, 2014. ACM.

[20] A. P. Felt, A. Ainslie, R. W. Reeder, S. Consolvo, S. Thyagaraja, A. Bettes, H. Harris, and J. Grimes. Improving SSL warnings: comprehension and adherence. In *Conference on Human Factors in Computing Systems*, pages 2893–2902. ACM, 2015.

[21] A. P. Felt, R. W. Reeder, H. Almuhimedi, and S. Consolvo. Experimenting at Scale with Google Chrome's SSL Warning. In *Conference on Human Factors in Computing Systems*, pages 2667–2670. ACM, 2014.

[22] M. Harbach, S. Fahl, P. Yakovleva, and M. Smith. Sorry, I don't get it: An analysis of warning message texts. In *Financial Cryptography and Data Security*, pages 94–111. Springer, 2013.

[23] J. Hodges, C. Jackson, and A. Barth. RFC 6797: HTTP Strict Transport Security (HSTS), 2012.

[24] R. Holz, J. Amann, O. Mehani, M. Wachs, and M. A. Kaafar. TLS in the wild: an Internet-wide analysis of TLS-based protocols for electronic communication. In *Network and Distributed System Security Symposium*. Internet Society, Feb. 2016.

[25] 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*, pages 427–444. ACM, 2011.

[26] L.-S. Huang, S. Adhikarla, D. Boneh, and C. Jackson. An Experimental Study of TLS Forward Secrecy Deployments. *Internet Computing, IEEE*, 18(6):43–51, 2014.

[27] R. Kang, L. Dabbish, N. Fruchter, and S. Kiesler. "My Data Just Goes Everywhere:" User Mental Models of the Internet and Implications for Privacy and Security. In *Symposium On Usable Privacy and Security (SOUPS)*, pages 39–52. USENIX Association, July 2015.

[28] M. Kranch and J. Bonneau. Upgrading HTTPS in Mid-Air: An Empirical Study of Strict Transport Security and Key Pinning. In *Network and Distributed System Security Symposium*. Internet Society, Feb. 2015.

[29] H. K. Lee, T. Malkin, and E. Nahum. Cryptographic Strength of SSL/TLS Servers: Current and Recent Practices. In *Internet Measurement Conference*, pages 83–92. ACM, Oct. 2007.

[30] G. F. Lyon. *Nmap network scanning: The official Nmap project guide to network discovery and security scanning*. Insecure, 2009.

[31] W. Mayer, A. Zauner, M. Schmiedecker, and M. Huber. No Need for Black Chambers: Testing TLS in the E-mail Ecosystem at Large. In *11th International Conference on Availability, Reliability and Security (ARES)*, pages 10–20. IEEE, 2016.

[32] B. Möller, T. Duong, and K. Kotowicz. This POODLE Bites: Exploiting The SSL 3.0 Fallback. *Google, Sep*, 2014.

[33] M. Oltrogge, Y. Acar, S. Dechand, M. Smith, and S. Fahl. To Pin or Not to Pin—Helping App Developers Bullet Proof Their TLS Connections. In *USENIX Security Symposium*, pages 239–254. USENIX Association, Aug. 2015.

[34] J. Payne, G. Jenkinson, F. Stajano, M. A. Sasse, and M. Spencer. Responsibility and Tangible Security: Towards a Theory of User Acceptance of Security Tokens. *arXiv preprint arXiv:1605.03478*, 2016.

[35] E. M. Redmiles, A. R. Malone, and M. L. Mazurek. I Think They're Trying to Tell Me Something: Advice Sources and Selection for Digital Security. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 272–288, May 2016.

[36] I. Ristic. Internet SSL survey 2010. *Black Hat USA*, 3, 2010.

[37] Y. Sheffel, R. Holz, and P. Saint-Andre. Summarizing Known Attacks on Transport Layer Security (TLS) and Datagram TLS(DTLS). RFC 7457 (Proposed Standard), 2015.

[38] Y. Sheffer, R. Holz, and P. Saint-Andre. Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS). RFC 7525 (Proposed Standard), 2015.

[39] E. Stobert and R. Biddle. The Password Life Cycle: User Behaviour in Managing Passwords. In *Symposium On Usable Privacy and Security (SOUPS)*, pages 243–255. USENIX Association, July 2014.

[40] S. C. Sundaramurthy, A. G. Bardas, J. Case, X. Ou, M. Wesch, J. McHugh, and S. R. Rajagopalan. A Human Capital Model for Mitigating Security Analyst Burnout. In *Symposium On Usable Privacy and Security (SOUPS)*, pages 347–359. USENIX Association, 2015.

[41] J. Sunshine, S. Egelman, H. Almuhimedi, N. Atri, and L. F. Cranor. Crying Wolf: An Empirical Study of SSL Warning Effectiveness. In *USENIX Security Symposium*, pages 399–416. USENIX Association, 2009.

[42] B. VanderSloot, J. Amann, M. Bernhard, Z. Durumeric, M. Bailey, and J. A. Halderman. Towards a Complete View of the Certificate Ecosystem. In *Internet Measurement Conference*, pages 543–549. ACM, 2016.

[43] S. Weber, M. Harbach, and M. Smith. Participatory Design for Security-Related User Interfaces. In *USEC*. Internet Society, Feb. 2015.

[44] K. Yakdan, S. Dechand, E. Gerhards-Padilla, and M. Smith. Helping Johnny to Analyze Malware: A Usability-Optimized Decompiler and Malware Analysis User Study. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 158–177, May 2016.

A Appendix

Recruitment Questionnaire

- Which of the following directives is used to host two different websites (www.website1.com and www.website2.com) within the same Apache webserver?

  - NamedHost
  - WebRoot
  - VirtualHost
  - ServerRoot

- Certificate files are usualy located at?

  - /root/ssl/certs
  - /etc/ssl/certs
  - /tmp/certs
  - /var/www/static/certs

- CSR means ...

  - common-name signing request
  - comodo signing request
  - certificate signing request
  - cross-site request

- Which is the best file permission for your private keys on a Linux system?

  - 0777
  - 0300
  - 0664
  - 0600

- Which command is used to find out the currently used IPs?

  - ifconfig
  - netstat
  - ipconfig
  - iptables

- Which files can the user www-data read?

  - -rw——— root root filname
  - -rw——— www www-data filename
  - -rwxrwxrwx root root filename
  - -rw-rw—- root www-data filename

- Which command is used to switch the user in Linux?

  - sudo
  - su
  - root
  - switchuser

- A symlink is created with which command?

  - ls -s TARGET LINK_NAME
  - symlink TARGET LINK_NAME
  - ln -s TARGET LINK_NAME
  - ln TARGET LINK_NAME

- TLS uses ...

  - symmetric cryptography
  - asymmetric cryptography
  - pem/der certificates
  - X.509

- TLS is ...

  - computationally very expensive
  - complex to configure correctly
  - originally invented by Facebook
  - easy to buy using cloud services

- Which of the following commands is used to save a file in vim (Vi Improved)?

  - Strg + S
  - Strg + X
  - Esc; :s
  - Esc; :w

- Which commands restarts the webserver?

  - sudo service apache2 restart
  - sudo /etc/init.d/apache2 restart
  - sudo service webserver restart
  - sudo service IIS restart

- The webserver has to have access to?

  - The private key used for TLS
  - The certificate used for TLS
  - The certificate authority private key for TLS
  - The certificate signing request used for TLS

- Where are HTML files served by the Apache Webserver located after default installation?

  - /usr/share/nginx/www
  - /etc/www
  - /var/www
  - /home/www

Lab Study Assignment

You are the system administrator at a SME (small and medium-sized enterprise). Your company runs a web portal and your boss instructed you to secure the communication by using TLS. Unfortunately you only have a very limited amount of time because your company will also soon be under security audit. This is why you should start right away deploying TLS. Make your configuration as secure as possible.
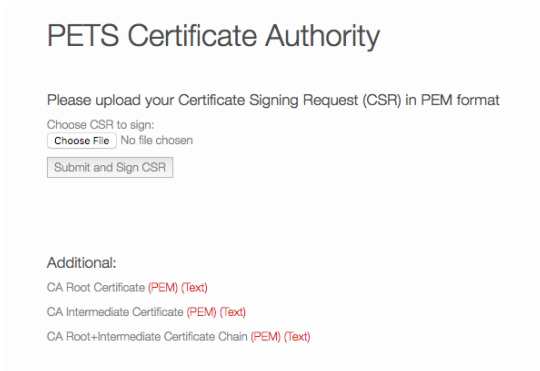
PETS Certificate Authority

Please upload your Certificate Signing Request (CSR) in PEM format
Choose CSR to sign:
Choose File No file chosen
Submit and Sign CSR

Additional:
CA Root Certificate (PEM) (Text)
CA Intermediate Certificate (PEM) (Text)
CA Root+Intermediate Certificate Chain (PEM) (Text)

Figure 2: Screenshot of the CA we implemented for the lab experiments.

System Configuration

- The company's web server (Apache2) is currently found at *http://web.local* on Port 80. There is only HTTP activated. No TLS configuration is made so far.
- You can connect to the web server with the command *ssh web* The username is *pi*, the password is *raspberry*. There is no root password, so you can just use *sudo* to execute commans as root user.
- You will have to use a Certificate Authority. You find a CA at *https://ca.local* Your client's Firefox trusts this CA called *TLS Userstudy Root CA*. You can test the certificate validation with this browser. The DNS names of both servers are locally configured at your client.

Post Lab Study Questionnaire

Demgraphics

- Participant ID (assigned prior to the lab experiments)
- Age
- Gender
- Months of industry experience

Experience with TLS

- Are you currently in charge of a web server? (Yes, I'm currently administrating a company web server./ Yes, I'm currently administrating a private web server./ Yes, I'm currently administrating at a profit/non-profit association. /No.)
- Have you ever installed and configured SSL/TLS before? (yes/no)
- Have you ever worked as a system administrator before? (yes/no)

Reflections on the Study Task

- Did you finish the TLS installation in the given time? (yes, no, I'm not sure)
- If you didn't finish the TLS installation in the given time, which steps are still missing to secure the communication? (open text)
- How difficult did you find TLS deployment? (Likert scale: very easy to very difficult)
- What did you find particularly difficult? (open text)
- What do you think are the key usability pitfalls of TLS deployment? (open text)
- What would you recommend a system administrator who has to deploy TLS? (open text)
- Is there anything else you would like to let us know? (open text)

Interview Questions - Expert Interviews

- As an auditor, how do you usually proceed to evaluate the security of a TLS configuration?
- What are the main vulnerabilities/configuration mistakes that you encounter as an auditor?
- What botheres admins/CSOs the most regarding TLS?
- What are the most critical steps in TLS deployment?
- How should the deployment process be improved?
- What piece of advice would you generally give to anyone in charge of securing communication over HTTPS?

Detailed Evaluation Criteria

Grade The overall grade for the configuration with a valid certificate. The grade is calculated based on the grading scheme from [2]. The score is based on individual ratings for protocol support (30%), key exchange (30%) and cipher strength (40%). The grade is issued based on the following cumulative scores:

- A: score $>= 80$
- B: score $>= 65$
- C: score $>= 50$
- D: score $>= 35$
- E: score $>= 20$
- F: score < 20

Errors/warnings/highlights. This refers to remarks that impacted the overall grading. The detailed description of these justifications is shown in Table 3.

Cipher strength score. This is represented by a number between 0 and 100, with 100 being the best possible. The cipher strength score contributes 40% to the overall grade. As weak symmetric ciphers can be easily broken by attackers, it is essential to the overall configuration that strong ciphers are used. SSL Labs evaluate ciphers based on an average cipher between the strongest and weakest. The scores are rated as follows:
0 bits (no encryption): 0
< 128 bits (e.g., 40, 56): 20
< 256 bits (e.g., 128, 168): 80
>= 256: 100

Key exchange score. As described in [2], the key exchange phase serves two functions: (1) authentication to verify the identity of the other party and (2) safe generation and exchange of secret keys to be used for the remaining session. Also, exportable key exchanges where only a part of the key is exchanged can make the session keys easier to compromise. Key exchange without authentication is vulnerable to MITM attacks and allows an attacker to gain access to the communication channel. Furthermore, the strength of the server's private key is crucial. The stronger it is, the more difficult it is to break the key exchange phase. Some servers use the private key just for authentication and not for the key exchange mechanism. Popular algorithms are the Diffie-Hellman key exchange (DHE) and its elliptic curve version (ECDHE). As in [2], the rating is calculated as follows:
Weak key or anonymous key exchange (e.g., Anonymous Diffie-Hellman): 0
Key or DH parameter strength < 512 bits: 20
Exportable key exchange limited to 512 bits: 40
Key or DH parameter strength < 1024 bits: 40
Key or DH parameter strength < 2048 bits: 80
Key or DH parameter strength < 4096 bits: 90
Key or DH parameter strength >= 4096 bits: 100

Protocol support score [2]. Several (older) versions of TLS have known weaknesses or are vulnerable to well-known attacks. The configuration is graded as follows with respect to the activated TLS versions. Again, if multiple versions are supported, the average between the best and worst protocol score is considered.
SSL 2.0: 0
SSL 3.0: 80
TLS 1.0: 90
TLS 1.1: 95
TLS 1.2: 100

Common name. This refers to the common name field specified in the CSR which specifies a FQDN (and re-spective subdomains if applicable) the certificate is issued for.

Key size. This refers to the size of the server's key pair.

Certificate chain length. This refers to the length of the certificate chain, including the server's certificate and certificates of intermediate CAs, and the certificate of a root CA trusted by all parties in the chain. Every intermediate CA in the chain holds a certificate issued by the CA one level above it in the trust hierarchy. In our example, the ideal length is 3.

Used provided CA to sign. In order to remove the bias from different CAs with varying usability, we implemented our own CA and provided the link to this CA in the assignment. Two participants did not use this CA and generated self-signed certificates instead.

Encrypted private key indicates whether the server's private key was encrypted by the study participant.

SSL 2 – TLS 1.2 indicates which protocol versions are supported.

RC4 support. To date, RC4 is considered weak and should therefore not be supported, unless required for compatibility reasons as found in [10].

Vulnerable to POODLE indicates whether the configuration is vulnerable to POODLE [32].

Forward secrecy indicates whether the configuration supports ciphers with forward secrecy (e.g., ECDHE).

HSTS indicates whether *HTTP Strict Transport Security* is configured. The security benefit of HSTS is that it forces secure communication with websites that use it by automatically converting all plain text and disabling click-through certificate warnings. If a client does not support HSTS, it simply ignores the header. Hence, activating HSTS enhances security with minimal effort without impact on compatibility.

HPKP indicates whether *Public Key Pinning* is used, which is a useful feature to prevent attacks and making the public aware of them.