RHEINISCHE
FRIEDRICH-WILHELMS-UNIVERSITÄT BONN

DOCTORAL THESIS

Behavioral Studies with IT-Administrators - Updating in Complex Environments and Securing Web Servers

---

Author:
Christian TIEFENAU

Supervisor:
Prof. Dr. Matthew SMITH

A thesis submitted in fulfillment of the requirements for the degree of Doctor rerum naturalium

Behavioral Security-Group
Institute of Computer Science IV

March 2021

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Rheinischen Friedrich-Wilhelms-Universität Bonn.

Erstgutachter: Prof. Dr. Matthew Smith
Zweitgutachterin: Dr. Katharina Krombholz

Tag der Promition: 17. März 2021

Erscheinungsjahr: 2021

Acknowledgements

The work presented in this thesis would not have been possible with the support of all the persons I met, collaborated and spent time with during the last five years: my advisors, co-authors, colleagues, research assistants, students, friends, and family.

First of all, I would like to thank my parents Achim and Astrid for their vital support throughout my life. Without them and their kind way of supporting their children, this work would not have been possible. Thank you also to my sister Andrea and my friends, who brighten up my life and inspire me in all kind of ways. Thank you, Roy, for being my friend for more than half of my life, Achim, Norbert, and all the other people, who call me Lothar, for their company throughout the last years.

I also want to thank my advisor Matthew Smith, who guided me in the last years and equipped me with the required tools to conduct good research. Especially in the time before deadlines, he motivated me with his seemingly endless knowledge and positive attitude.

I am grateful that I was able to collaborate with a lot of great persons over the last years. First and foremost, Maximilian Häring. I really enjoyed our weekly boulder sessions. Eva, for being there for me and supporting me in the final months of writing this thesis. Emanuel von Zezschwitz, who was my supervisor for one year throughout this thesis, and Katharina Krombholz, who I appreciate for having one of the most positive attitudes I know. I also want to thank Karoline Busse, Alena Naiakshina, Anastasia Danilova, Ronald Brenner, Sarah Prange, Florian Alt, Mohamed Khamis, Marco Herzog, and Sergej Dechand. I enjoyed working together with all of you.

Also, I want to thank all my colleagues of the Behavioral Security Group I missed in the list above. I enjoyed spending much time in discussions at lunch and having fun in struggling with statistics.

Bonn, March 2021

RHEINISCHE FRIEDRICH-WILHELMS-UNIVERSITÄT BONN

Abstract

The Faculty of Mathematics and Natural Sciences
Institute of Computer Science IV

Doctor rerum naturalium

Behavioral Studies with IT-Administrators - Updating in Complex Environments and Securing Web Servers

by Christian TIEFENAU

Up until the turn of the millennium, research in the field of IT security mainly focused on the technical aspects of security mechanisms. Since then, the human factor has become more and more important and sparked research in the very broad field of usable security and privacy. In this field, researchers study the human-aspects of security systems, such as understanding security mechanisms and user-behavior when it comes to picking passwords or updating their systems. While these works mainly focused on end users, recently, expert users have become the subject of research as well. In understanding developers and administrators, we can identify problems they face in performing security-relevant tasks and developing systems that support them, resulting in enhanced system security. This thesis extends the field of usable security research and presents the results of four studies involving IT-administrators and expert users, which focus on the update processes in corporate contexts and the TLS setup step in the web server configuration. The first study analyzes the update process of administrators in companies. This study also reveals obstacles that occur at various points in this process, which can be a reason for delaying or not deploying updates. Based on the emerged process model, I further present a case study in which I apply the model to update processes of a web development company. The results show that the process is far more flexible than originally thought, leading to an adapted version of this model. Subsequently, I present the findings of a study related to the importance of specific components in update release notes. The findings of these three studies serve as a foundation to spark future work, e.g., in researching better communication strategies of the changes an update brings or finding ways to reduce the delay of updates by preventing downtimes. Following the update topic, I present a study on the analysis of the automation effect in the TLS configuration process. The automated approach was found to have a positive impact on the security of the configuration. Through this study, I present lessons learned and discuss areas where the automated approach's principles can further enable better usability and security in the context of IT-administration.

Contents

Dedicated to my family and friends.

Chapter 1

Introduction

For a long time, IT security and research mainly focused on the technical background of technologies. Starting with the work of Adams and Sasse's "Users are not the Enemy" [7] and Tygar and Whitten's "Why Johnny can't Encrypt" [140] in 1999, a whole new field of research emerged, focusing on the human-aspects as well. As part of the human-computer interaction field, Usable Security and Privacy began to get more attention; thus the number of publications in this field has grown from year to year. With the advent of the Symposium on Usable Privacy and Security (SOUPS), it even has its own conference. The motivation of this research field is clear: If we understand how humans think about and work with technology, we can improve the software and products we design, including the practical security and privacy that come with them. In the beginning, most of the research in this area focused on end users, by, for example, observing their update behavior [132, 131, 136], their understanding of email encryption [140], or security warnings [119], amongst other important topics. Over time, it became clear that not only end users are the cause of security incidents, but other stakeholders (i.e., developers and administrators) can also influence the security of systems. Taking "Developers are not the enemy" by Green and Smith as an example, the research community proposed the extension of usable security to study these stakeholders as well [59]. In such studies, researchers observed the usability of different cryptographic APIs for developers [2] or why developers are struggling in terms of storing passwords securely in a database [98, 96, 97], for instance.

It is important to understand experts' problems and mental models, as their decisions and actions can have impacts on a large number of systems and/or users. A security topic in this field can be observed through the lens of different stakeholders, with every one of them dealing with different problems. Taking Transport Layer Security (TLS) as an example, studies observed its implementation on the client-side in Android apps. They found that some developers bypass the security mechanism by allowing all certificates or miss an understanding of features like pinning, which exist to improve security [46, 101]. On the server-side, research found that the correct deployment of TLS configurations can be hard because of the complex workflow that must be understood regarding multiple security-related topics like encryption or key-algorithms [79].

While developers have been the subjects in a growing number of studies, the focus of this work is on administrators. In the first section of this work, I will present findings on the struggles administrators face and approaches they take to mitigate them in the context of updating. Updating software and systems is an important security measure that experts agree on [71, 110, 25]. It is easy to improve the security of systems by applying updates, so they are hardened against vulnerabilities like Heartbleed [38]. However, many systems in the wild remain vulnerable for two years or more [120], and even in July 2019, more than 90,000 machines had not been patched [116]. The Equifax breach in 2017 is a viral example of a situation where a deployed update would have prevented a security breach and the leakage of the data of more than 145 million people [68].

It is essential to understand how end users and expert users are handling updates, so we can understand their struggles and help them, for example, by shortening the time between an update release and its deployment. The usable security community already observed the update processes and behaviors of end users in numerous studies [132, 107, 52, 94, 133, 136, 42]. However, update processes in a corporate context are far less understood. Here, related work has found that, in this context, other factors like business needs drive decisions to update [93], and security professionals prioritize security aspects over potential usability consequences [133]. This work contributes to this growing body of knowledge by studying and understanding the administrators' behaviors, experiences, and attitudes regarding updates in a corporate environment to act as a starting point for further investigation. In chapter 3, I first observed the update topic for administrators based on the results of an interview study and subsequent online survey. Out of those, I quantified common practices, presenting an update process model and obstacles (e.g., downtime or lack of information about updates). The findings indicate that even experienced administrators struggle with update processes, as the consequences of an update are sometimes hard to assess. Based on this knowledge, in chapter 4, I present a case study conducted in a web development company where I applied the proposed model and that of a related work. In this study, one researcher was embedded in a company that handled updates of their customers' web content management systems for one year. This allowed an analysis of the company structure and internal tickets covering ten years of information concerning the update processes that the staff followed. This in-depth view of update processes showed that for this case study, the proposed update process-models, while being helpful, were not sufficient to model the uncovered processes. Out of these findings emerged an improved and more flexible update process-model that better represents the process. The results of these studies required a high level of abstraction due to the unique setting of each administrator and the complex task of updating itself in these settings. This makes a comparison and generalization within different environments difficult to impossible, but can serve as a foundation for further research that focuses on different aspects of the process or well-defined scenarios. In order to support administrators in executing security-relevant tasks like

updating, we need to zoom in from this broader view to a specific task in the process. One example is presented in chapter 5. A large part of the update process is the "information" and "deciding" stage, where administrators gather information about the update. This helps them to foresee its impact by reading release notes that the vendor provides and that can contain information about the version, release date, and fixes or changes that come with each patch. In this chapter, I present findings on the importance of the contents of update release notes.

The focus on a specific task can also be used to research ways to support administrators in contexts other than updating. When administrating web servers, administrators face the task of configuring TLS to enable a secure communication between the web server and its clients.

Like the update topic before, TLS has been an active research topic in the usable security domain, especially regarding end user's perspective [119, 49, 111]. But again, it is important to take a look at the persons that are "on the other/server side," who are responsible for the web server communication, since research has shown that end users would see 15,400 false positive warnings per true positive warning due to server misconfigurations [11]. For a long time, the TLS configuration task had to be done manually, but with the come up of Let's Encrypt (LE) in 2015 and Electronic Frontier Foundation's (EFF) tool, Certbot, it is now possible to automate the acquisition and configuration of LE certificates for web servers [41]. In the final chapter 7, I present a study about the task of TLS configuration that administrators have to deal with in the web server-context. The conducted experiment observes the impact of the automation that Certbot offers on TLS deployment and the security of the configuration. Using a within-subjects lab study design, the results show that usability improvements like automation can significantly impact security and should be considered in other security-related tasks that experts struggle with in order to lower the complexity.

All chapters are based on previously published or currently under review work. Therefore, there is a disclaimer at the beginning of each Chapter stating my contributions and those from my co-authors to each work.

Chapter 2

Related Work on Updates

This chapter contains the related work that is relevant for the presented update-related studies in this thesis in chapter 3 to chapter 5. These studies are (1) about the update behaviour of users and (2) about investigating the security behavior of expert users. Following this, a paragraph about web content management systems (WCMS) should provide information about the distribution of WCMS in the web to give a context for the study in chapter 4.

2.1 Users' Update Behavior

According to security experts, keeping systems and software up to date is an important security recommendation [109]. However, users may not follow this advice for reasons that are not related to security [107], and only a minority of non-experts actually considers software updates an important security measure [71, 99]. It has been repeatedly shown that users often delay or even avoid updates [52, 94, 133].

Investigation of the root causes of such critical user behavior has become a very active field of research. Previous work revealed diverse reasons for avoiding updates. Many users think that updates are not important because the link to security aspects often is not obvious [42, 55, 90, 106, 136, 132, 133]. Furthermore, users are often afraid of functional changes (e.g., UI modifications) [18, 132, 131, 133] or fear making mistakes [52]. Inconvenience is an important factor as updates can cause interruptions and take time [90, 136, 133]. Finally, bad experiences with previous updates and negative online reviews hinder the installation of future patches [42, 90, 123, 131]. This problem seems self-perpetuating, because the frequency of security updates is influenced by the emergence of novel attacks and thus, cannot be controlled by the vendor alone [114]. However, high update frequencies can lead to further negative reviews [51, 105].

Several countermeasures for mitigating the problem of delayed updates have been proposed. As one straightforward solution, automatic updates [136] and silent updates [34, 114] have been deployed. Although such mechanisms are very effective in keeping software up to date, they often cause confusion and irritation as they hamper the user's understanding of what is happening on their machines [39, 136]. Furthermore, some users might have good reasons to refrain

from performing certain updates [39]. Therefore, user-centered solutions, such as providing more information [91, 103, 123, 122] and designing better notifications [43, 44, 54], have been repeatedly suggested as complementary concepts to further increase compliance rates.

2.2 IT Professionals and IT Security

Recently, researchers have started focusing on security-related usability problems of specific user groups [3]. In contrast to security advocates [62] or security analysts [58], most of these people are not security professionals. They are often knowledgeable in a specific domain, related to IT. Several recent studies addressed the problems of software developers [6, 4, 14, 82]. For example, Acar et al. [6, 4] investigated available sources of information and how these sources influence code security. Gorski et al. [82] showed that software developers benefit from API-integrated security recommendations.

Several human-centered studies with system administrators were published between 2001 and 2007. In 2001, Hrebec and Stiber [70] studied the mental models of system administrators and found that these experts often struggle to understand the complex systems that they need to manage. In addition, the study participants reported a lack of formal education and the desire to solve problems by themselves. Barrett et al. [17] found that system administrators often lack situational awareness. Haber and Kandogan [61, 74] and Botta et al. [22] observed the tools and work practices of security administrations and IT professionals. Their results show that security administrators perform a lot of different tasks and need various skills like pattern recognition or inferential analysis to perform these tasks. They proposed, that new classes of tools need to be developed to counter the ever increasing complexity of the systems and attack-vectors.

In contrast to this early work, a few recently published papers investigated more specific problems of system administrators. Fahl et al. [45] studied non-validating X.509 certificates and revealed that about 30% of the responsible webmasters misconfigured their web servers accidentally. Ukrop et al. [128] analyzed the corresponding warnings and found that rewording can help administrators to make better informed decisions. Krombholz et al. [79, 80] showed that the deployment process for HTTPS is far too complex and that administrators struggle with finding secure and compatible configurations due to the lack of conceptual mental models. Dietrich et al. [32] investigated the administrators' general perception of misconfigurations and identified missing or delayed updates as one of the root causes of these problems.

There exists work that discussed update processes in companies [20, 21, 93, 133]. For example, Vitale et al. [133] performed three interviews with technical staff concerned with updates and found that these professionals prioritized security aspects and licensing issues over potential usability consequences. This finding confirmed previous findings [93] that in a corporate context, business

needs rather than user requirements drive update decisions. In contrast, Blythe et al. [21] reported that employees often rely on "security experts" in the company to manage updates and often lack a feeling of responsibility. Finally, the update challenges of system administrators have been indirectly considered by various researchers who proposed automatic tools to improve the manageability of the update process (e.g., [16, 56, 81, 100]). However, none of these concepts have been evaluated in a user study.

2.3 Web Content Management Systems

In the context of the presented study in chapter 4, most of the observed software were web content management systems (WCMS). WCMS are technical systems that support the maintenance, presentation, organization, and use of processed information (i.e., text, photos, videos). Using graphical user interfaces, information and metadata can be processed into a web format without requiring in-depth technical knowledge [92]. This helps to efficiently maintain the content for websites of small- and medium-sized businesses [115].

Generally, a WCMS is structured as a server-client based system, where the content- and administration management components lie server-side. Attached to these are services to transform the content into several output formats, for example, a desktop or a mobile version of a website. The content and meta information are usually saved in a database, and through given tools on the client-side, the user can access the components of the content server [115]. According to the W3C, the WCMS WordPress [141] is used in 38.5% of all websites on the internet [129]. In Germany, web content management systems like Joomla! [73] and TYPO3 [126] are the most represented systems besides WordPress. Together they hold an aggregated market share of 70.78% [121]. Many WCMS simplify the update process of their backend through a wizard or by providing automated background updates. However, it is also possible to manually update the system by modifying specific files. Similar to modern software architecture, the available functions of the WCMS can be expanded through plugins (also called modules or extensions). The update of installed extensions is usually done through an extension manager on the web interface. Manual installation is also possible albeit more laborious [69]. Some of the most popular plugins show more than 5 million installations [142]. Because they are widespread, unsafe plugins are an attack vector to consider, alongside bugs and a system's incorrect configuration [115]. A single unsafe plugin allows attackers to apply the exploitation of a vulnerability upon a multitude of websites, as was the case with the Profile Builder and Profile Builder Pro (< version 3.1.1) plugin for WordPress, where a vulnerability allowed unprivileged users to gain administrator rights. It is estimated that 65.000 websites were affected through the installation of this plugin [29]. Attackers could then use these compromised sites to distribute malware and spam, malicious redirects, or merely the defilement of the site [139].

When looking at the period from 2015 to 2018, WordPress released 48 updates out of which four were fixes for high-risk Common Vulnerability and Exposures (CVE)s with a CVSS score greater than 7 [27]. TYPO3 released 40 patches, and out of these, no high scored CVE in this period [127]. For Joomla!, there were 36 patches in which eleven high-risk CVEs were fixed [30].

Chapter 3

Exploring Update Behavior of System Administrators

Disclaimer

The contents of this chapter were previously published as part of the paper "Security, Availability, and Multiple Information Sources: Exploring Update Behavior of System Administrators" presented at the 16th Symposium On Usable Privacy and Security (SOUPS) in 2020 [125] together with my co-authors Maximilian Häring, Katharina Krombholz, and Emanuel von Zezschwitz. As this work was conducted with my co-authors as a team, this chapter will use the academic "we" to mirror this fact. The idea and initial concept for this work came from me. As it was part of his Master Thesis, the user-studies were designed by Maximilian Häring and me. Maximilian Häring and Karoline Busse conducted and transcribed the interviews. Maximilian Häring and I coded and analyzed the interviews. Together with Katharina Krombholz and Emanuel von Zezschwitz, we created the key observations based on which Maximilian Häring and I created the survey. I analyzed the quantitative part. Before compiling the paper for publication, Emanuel von Zezschwitz, Maximilian Häring, Katharina Krombholz, and I jointly discussed the study's implications.

3.1 Motivation

"Keep your systems up to date" is one of the most popular pieces of advice that security experts give to end users [71, 109]. Supporting this, Khan et al. found that there is a correlation between not deployed updates and infected machines [75]. Systems can easily be hardened against vulnerabilities like Heartbleed [1] by applying updates. Regardless of that, many systems in the wild remain vulnerable for two years or more [120]. A prominent example of a situation where an update could have prevented severe data leakage is the Equifax breach [2],

[1] `http://heartbleed.com/`, accessed 02/25/2020.

[2] `https://www.theverge.com/2017/10/3/16410806/equifax-ceo-blame-breach-patch-congress-testimony`, accessed: 11/20/2019.

which occurred in 2017. Similar incidents seem not unusual as is reported by an industry report [86].

Related work studied user perceptions and experiences with system updates and found that the results are often not in line with current recommendations of experts from a security perspective. In most cases, concerns about functional issues or unexpected UI changes hinder individuals from updating their systems [132]. In addition, users often do not understand the importance of non-visual changes [132], as they come with security updates. In contrast to users who are responsible only for managing their own personal devices, system administrators are in charge of large and complex IT infrastructures while also being users. We argue that their update behavior can have severe implications at a much larger scale. Marconato et al. [88] observed the vulnerability life-cycle on different platforms and found that the time to patch and disclose vulnerabilities is decreasing. This finding can be applied to the Equifax breach and suggests that administrators are required to react in a timely manner.

Although general user concerns about system updates have been investigated in user studies, little light has been shed on the perspective of specific user groups (e.g., administrators or operators). Investigating administrators, Dietrich et al. [32] found that insecure configurations are often caused by institutional and individual factors, as well as time constraints. We assume that similar factors can have a negative impact on update processes. Administrators are often overworked [32], and updates are time-consuming. Secure systems, however, rely on updates and therefore, require regular attention by administrators. As the body of literature is still in an early state regarding administrators' update behavior, we follow an inductive approach to explore the processes and obstacles that administrators face when updating in a corporate context.

Our contributions are as follows:

- We conducted **seven qualitative interviews to explore how administrators experience, perceive, and act during the update process**.
- We conducted an **online survey with 67 valid answer sets to test our observations on a larger scale.**
- We confirm that current **update processes and system factors tend to endanger IT security** and we discuss critical factors that need to be addressed to **support administrators**.

Parallel to this work, Li et al. [84] published a closely related paper in which they studied US-based system administrators in a qualitative fashion. They as well researched the update process in companies and found several *pain points* within the process. In contrast, the interview sample of this work was drawn from German companies, thus representing a different culture. Overall, the study presented here confirms most of their findings. We will separately discuss our findings in comparison to Li et al.'s in subsection 3.4.5 in more detail.

The related work to this chapter can be found in chapter 2.

3.2 Interview Study

Although recommendations for patch management have been published[3], we are aware of only one other study that systematically investigated the update behavior of system administrators [84]. Therefore, we started with an interview study to identify important factors of the problem space.

This interview study aimed to provide answers to the following research questions with an emphasis on administrators' perceptions, challenges, and tools they use in their update routines:

1. **How can the update processes be described, and what common patterns are there?**
   Administrators are usually paid professionals who are responsible for updating large and complex IT infrastructures. This raises the question, whether, and if so, where, system administrators' updates processes differ from end users' processes [131].

2. **What issues and obstacles do professional administrators face in their update routines?**
   We specifically aim at understanding the problems of administrators and their perception of update processes. Identifying obstacles in relation to processes, tools, and environments is indispensable to define important directions for future work.

3. **How are administrators informed about updates, and which sources of information do they use?**
   Related work has indicated that the source of information can have a significant impact on software security [6, 50]. Thus, we aim at understanding how administrators gather information and what sources they use.

4. **What kind of tools do administrators use to manage system updates, and is there room for improvements?**
   As usable security researchers, we are specifically interested in the tools involved in the update process. We hypothesize that although some tools are used on purpose and other tools are unavoidable, such tools can either complicate or ease the process.

3.2.1 Study Design and Procedure

We conducted seven semi-structured interviews in June 2018 to explore the participants' opinions, thoughts, and experiences. Based on three pilot-study interviews, we refined the interview guidelines to balance between informing the research questions and supporting a flexible exploration of the problem space

[3] `https://www.infosec.gov.hk/english/technical/files/patch.pdf`, accessed 02/25/2020.

Pseud.	Position/Task	Age	Exp. (Years)	Team size	Supervised Machines
Markus	Administrator	25–35	6	7	300–350 clients, 150 virt. servers
Lorenz	Update management	25–35	2	n/a	5 servers
Cyril	Administrator	25–35	6	15	10,000 virtual, ca. 100 physical
Milan	Help desk	25–35	2.5	12	600 clients, number of servers
Zelko	Administrator	25–35	10	2	16 physical, 35 virtual, 80 clients
Alex	Update management	> 35	23	5	26 physical, 170 instances
Julian	Management	> 35	29	20	n/a

TABLE 3.1: Interview participants.

(i.e., leaving enough room to add further comments). The interview was structured into (1) general questions about the daily work routine of the participant, (2) general experiences with updates, (3) a more detailed assessment of specific aspects, and (4) additional comments. The guidelines are in section A.2.

All but one interview were conducted by the same researcher. Both researchers are experts in computer science and spoke the same native language as the interviewees. After an introduction to the purpose of the study, the participants were asked to sign a consent form. All participants gave their consent to being audio-recorded. We conducted one interview in person and six via telephone. All interviews were held in German. During the interviews, the interviewee and the researcher were allowed to take notes. The interviews lasted between 34 and 67 minutes and ended with a short questionnaire that collected demographic information.

3.2.2 Recruitment and Participants

We did not restrict our invitations to administrators working with a specific operating system, infrastructure or type of update. The only criterion for inclusion was that participants had to be in charge of, or in contact with, any kind of updates. Personal contacts were used as entry points to larger organizations and asked to forward the announcement to their employers' IT department. Additionally, we directly approached representatives of medium-sized and large companies at CeBIT 2018, a large international computer expo[4].

In total, we recruited seven participants at companies that had an office based in Germany. All participants reported they were in charge of system administration, although they had various job descriptions and managed different types of systems. Table 3.1 presents more details about the sample. All the participants were male. For ease of readability in the following sections, we assigned the participants random names.

[4] `https://www.cebit.de/`, accessed 02/25/2020.

3.2.3 Analysis

The interviews were transcribed, and coded by two researchers. We coded open answers inductively following the approach of Wertz, Charmaz et al. [138]. The two researchers categorized the data according to the research questions presented in section 3.2. The first three interviews were coded in a batch to establish the first codebook. Each of the following four interviews was coded separately. Then, the conflicts were discussed, and new codes were added to the codebook. We calculated the combined Krippendorff's alpha [78] before (0.61) and after (0.98) the discussion phase for each interview. Our goal was to use the qualitative analysis solely as a first step and foundation for the following quantitative study. Therefore, we refrained from continuing with interviews until theoretical saturation [53] was reached.

3.2.4 Qualitative Results

In the following, we present the results from the interview study with respect to the research questions.

Update Processes

In Table 3.2, we present the sum of all extracted process stages, including all reported steps that were performed in these stages. Overall, the update process varied in time and structure among participants and tended to be variable even for individual administrators, depending on the software that needed an update. Cyril reported he worked in a client environment with Windows systems. He was concerned mainly with regular update cycles. Therefore, he was able to prepare for update events (e.g., briefing the team, allocating resources, allocating maintenance windows, and gathering information). Four out of seven participants reported they relied on fixed update cycles for client systems, although Zelko reported that this was not always possible in practice. In contrast, Lorenz, who worked at a smaller company, reported that employees at his company were responsible for their systems. When we discussed more specific software, the answers became more diverse. Milan usually builds packages to automate the distribution, but Markus tends to perform manual installations.

Although participants' responsibilities differed, we were able to identify common patterns in the update process. Most of these stages can be mapped to those of client users [131]. However, we identified three major differences:

Some administrators perform extensive testing before installing the update on a live system. For example, Julian utilized up to three stages. Zelko, who stated, that "[E]ven if there is a risk that the update breaks something, we install them timely", utilized two test stages. First, he tested the update with virtual machines that simulate the client landscape, and then he rolled out the updates for a small group of colleagues.

Stage	Step	Obstacles
Information	Becoming aware	
	Further details	Unsatisfying communication with the publisher*
Deciding	Discussion	Stability (1); Risk of exploits (2); Performance (1); Priority (2); Missing expertise (1)
Preparation	Planning	Planning itself (3); Time of release (3); Communication (1); Missing documentation about the system and processes*
	Backup	
	Waiting for release	
	Obtaining the patch	Missing patches (1)
	Automating	
	Informing users	
Testing	Test system	Testing itself (1); Broken dependencies (4); Resources*; Frequency of updates*
	Pilot system	
	Problem solving w. vendor	
Installation	Installation itself	Failure (2); Missing configuration options (1); Social pressure; System resources (2); Complexity (3); Missing tools (3); Heterogeneous system (6); Company structure (3); Impact on systems/users (2); Downtime (1); Installation method (manual/automatic) (1,1)
	User interaction	Waiting for users (1)
	Reboot	Reboot itself (3); Old/Slow hardware (1)
Post-Installation	Documentation	
	Testing/Monitoring	
	Troubleshooting	
	Reversing	Missing backup, failover, or redundancy*

TABLE 3.2: Overview of stages, steps, and obstacles. The number in brackets denotes the number of participants who mentioned this aspect in the interviews. *Additional obstacles were found through the questionnaire.

Updates are rolled out step by step. The participants reported that often not all systems are updated in one batch. This allows the administrators to minimize the number of misconfigurations once an update fails, but constraints on resources are also a reason for this. For example, Julian reported that the network would be used to capacity if all systems were patched at the same time.

The preparation step is structured and involves planning and research of resources and the allocation of time slots. Five participants explicitly reported they conduct online research before they install an update. In addition, Alex told that important update decisions are often made in group discussions.

Obstacles

We identified various obstacles that hamper the administrators' task of performing updates. In Table 3.2, we connect and report obstacles to the stages of the update process. In the following, we discuss common obstacles in more detail:

Downtimes. The participants stated that downtimes are a serious obstacle in the update process which often cause delayed deployments. As soon as a reboot is necessary, and there is no redundant system, downtime is induced. Alex gave anecdotal evidence of a mitigation strategy: Upgrading from Solaris 10 (which required significant downtime) to Solaris 11 (which supports near to hot-swap updates and an easy rollback) increased update frequencies from three times a year to once a week.

Dependencies. The participants reported patches that break dependencies usually delay the process. Although this may not be surprising, it highlights the problem of dealing with dependent systems that cannot be patched in time. Further dependencies are introduced as part of the infrastructure landscape. For example, some systems depend on other systems to be available at boot time (Markus). Assessing these dependencies and then following the right order makes the process highly complex. Another type of dependency is towards the vendor of the software or hardware. An example of this can be as trivial as no available patches, even if a vulnerability is public, as Lorenz reported for the Meltdown case.

High frequency and large files. Every update takes resources: for example, time, workforce, CPU, and data storage. Zelko reported that big update files, which are often a consequence of combining functional updates with security patches, can cause problems. To handle resource constraints, updates are rolled out in multiple but smaller batches (Julian).

Competing priorities. Similar to standard users, administrators' decisions to perform updates are influenced by various factors. Participants reported stability considerations, the risk of an exploit, and performance issues as influential aspects. The fact that some systems do not separate security and feature updates may intensify this situation. Finally, required resources are sometimes allocated to other processes that have higher priority. Alex reported that "the decision [to update] is always based on the sum of available information". As mentioned in section 3.2.4, group discussions are an important part of the process. However, the need for communication can also delay updates (Milan).

Human Factors. In addition to technological and structural constraints, the administrator faces other obstacles. Missing expertise or a lack of knowledge can lead to situations where administrators rely on third parties. In this regard, Lorenz acknowledged that he does not always know how to act correctly. Or as Markus put it, he has to trust the vendor that the classification of the patch is correct. System administrators have to trust the information they get from the software developer, vendor, or other source. Another factor we identified is social pressure, as Lorenz reported, "And you look like an idiot, when you kill a

git server. [...] That chases me." Another aspect that makes updating harder for administrators was software which is managed by end users. Such software is often installed without the knowledge of administrators and makes the update process more complicated because it is not integrated in standard processes.

Sources of Information

The participants reported they use various methods to inform themselves about security updates and vulnerabilities. Five out of seven participants reported they use third-party sources that were independent of the software publisher, such as popular news portals or blogs. This information is usually supplemented by publisher-related newsletters and specific mailing lists, such as the Ubuntu-security mailing list (Lorenz). Cyril mentioned specialized third-party services that push information about available patches. Others got more specific and reported that they use tools like SCCM[5] or Nessus[6] which serve as sources of information.

Tools

The participants reported OS-integrated tools and special purpose tools that are used to update servers and clients and that serve as sources of information. The purpose of such tools ranged from monitoring systems (Julian) to complete automation of the update process, such as SCCM or WSUS[7] (Markus). Participants also named external services (e.g., Shavlik[8]) that test and pre-filter patches for companies. Although automation of update processes was an important goal for participants, it had not yet been fully implemented. Software that is not covered by such tools, meaning not integrated by default, has to be updated manually or integrated. This seems to be the case when the vendors or the operating systems differ (e.g., using Microsoft WSUS to update Adobe Flash Player). Although the integration is possible, it is connected to additional effort and is not always done (Markus), e.g., if it affects only a small group of clients (Milan). Concerning future developments, Lorenz was less optimistic and brought up that the time investment in tools that would ease the workflow was not a high priority.

ID	Observation
	Update Process and Information
U1	Online sources are an important source for administrators to get informed about updates.
U2	Small companies have no formal update process.
	Update Obstacles
O1	Performance considerations often hinder the installation of an update.
O2	Update-caused downtimes delay the installation of an update (e.g., reboots)
O3	Problems after the installation of an update on the live system are only a minor concern.
O4	Lack of information hinder the update process.
O5	User action (e.g., installing a software without the knowledge of the admin) can circumvent the update process and render it useless.
	Human Factors
P1	Administrators of big companies feel sufficiently trained.
P2	Administrators think that timely updates are important.

TABLE 3.3: Key observations based on qualitative results.

3.2.5 Key Observations

We performed an interview study of administrators' update behavior. Based on the research questions, we were able to describe update processes, common obstacles, information retrieval, and the use of software tools. We extract a series of key observations to guide the construction of the quantitative study, following the interviews. Table 3.2 provides an overview of the process stages and obstacles that administrators face in their daily lives according to the participants. Table 3.3 presents nine key observations, which were formulated based on the qualitative findings and then categorized in three groups: "Update Process and Information," "Update Obstacles," and "Human Factors." In the next section, we report on a quantitative online survey which was performed to shed further light on the update behavior of system administrators.

[5] `https://www.microsoft.com/en-us/cloud-platform/system-center-configuration-manager-features`, accessed 02/25/2020.

[6] `https://www.tenable.com/products/nessus/nessus-professional`, accessed 02/25/2020.

[7] `https://docs.microsoft.com/en-us/windows-server/administration/windows-server-update-services/get-started/windows-server-update-services-wsus`, accessed 02/25/2020.

[8] `https://www.ivanti.com/company/history/shavlik`, accessed 02/25/2020.

3.3 Quantitative Online Survey

Following the interviews, we performed a quantitative online survey. We created statements based on our observations in the interview study and developed an online questionnaire to quantify and enrich them.

3.3.1 Procedure and Structure

The recruitment process for the preliminary interview study indicated that system administrators are inherently short on time, and thus, minimizing the time to fill out the survey was indispensable to obtain a sufficient number of responses. Therefore, most of the questions were based on simple answer types, such as check boxes or rating scales. To further motivate participation, we offered an opt-in for a raffle of 3D prints. Every tenth participant had the chance to win a 3D-printed model of their choice. E-mail addresses were collected only for this raffle, stored separately, and deleted afterwards. Twenty-three entered their contact email address of whom no one was interested in a print. After participants had given their consent to take part in the study, the survey started. Completion took about ten minutes.

To support many different circumstances, we framed questions in a way that answers could be related to the current position or if not applicable, to the last position as system administrator. We started by collecting demographic data (e.g., age), personal information (e.g., years of experience), information about the work environment (e.g, their role, company size), and information about update processes (e.g., existence of formal processes). In the second phase, participants rated 1) the frequency of specific events using 5-point scales ranging from “1 - Never” to “5 - Always” and 2) indicated their agreement with different statements using 7-point scales (“1 - Strongly disagree” to “7 - Strongly agree”). The questions were presented in random order for each participant. The questions were chosen based on our observations and thus, examined the impact of obstacles (e.g., “Downtimes caused by the update process hinders the installation of an update”), human factors (e.g., “I feel that I am sufficiently trained as an administrator”), and information sources (e.g., selection of sources used). The questionnaire ended with an open-ended question about the biggest obstacles in the update process that we coded afterwards. The new categories are marked with an asterisk in Table 3.2.

To ensure the internal consistency of the collected data, we added an attention check based on the negation of one of these questions. Five participants, who answered both questions with a different polarity, were excluded from the evaluation.

Survey demographic data	
n	**67**
Gender	1 Female
	58 Male
	3 Other
	5 Not specified
Location	19 North America
	41 Europe
	7 Rest of the world
Age	**22 – 55**
Statistics	$md = 34, mn = 34.5, sd = 7.8$
Experience	**0.1– 30.0** years
Statistics	$md = 10.0, mn = 11.1, sd = 7.0$
Company	34 IT-related
	29 Non IT-related
	4 Other
Company Size	4 ≤ 10
	15 $10 < x \leq 50$
	15 $50 < x \leq 250$
	33 > 250
Role	50 Full-time admin
	11 Not primary, but $> 20\%$ of time
	6 Not primary, but $< 20\%$ of time
Administered Systems	28 Clients
	63 Servers
	14 Mobile
	13 Other

TABLE 3.4: Demographic data from the online survey.

3.3.2 Recruitment and Participants

To attract professional system administrators, we decided against using crowdsourcing platforms like Amazon Mechanical Turk. Instead, we reached out to community sites like Reddit and specialized forums. Additionally, we used Twitter and followed a similar approach as we did in the interview study. Posting in forums resulted in 66 answers, advertising on Twitter resulted in 67 responses, and using personal contacts in companies to spread the questionnaire contributed eight answers.

The English survey was active for 14 days in September 2018. During this time, the questionnaire was started 141 times and completed by 72 (51.1%) participants. As reported, five data sets were excluded from the analysis due to failed attention checks, resulting in 67 valid data sets. The participants' age ranged between 22 and 55 years. Fifty-eight of them were male, one female,

ID	Statement	1	2	3	4	5	*	Plot	Med.
O1	Performance considerations hinder the installation of an update.	24	27	7	9	0	0		2
O2	Downtimes caused by the update process hinder the installation of an update.	8	22	13	18	6	0		3
O4	A lack of information about the update hinder the installation of an update.	15	19	18	9	4	1		2
P1	I feel sufficiently trained as an administrator.	1	7	13	29	17	0		4

TABLE 3.5: Overview of the responses to statements regarding the frequency on a 5-point scale from "1 - Never" to "5 - Always" (* "Not sure") and their connection to the key observations.

ID	Statement	1	2	3	4	5	6	7	*	Plot	Med.
O3	Post-installation problems in a live system are only a minor concern because they don't happen frequently.	8	9	8	5	12	16	9	0		5
O5	Users often install software without the knowledge of the administrator.	18	9	7	8	12	6	7	0		3
P2	Deploying security updates in a timely manner is important.	0	1	0	0	7	18	41	0		7

TABLE 3.6: Overview of the responses to statements regarding the attitude on a 7-point scale from "1 - Strongly disagree" to "7 - Strongly agree" (* "Not sure") and their connection to the key observations.

three reported "Other" and five preferred did not specify their gender. More than 61% (41) work in European countries. The biggest group of the participants pool work in Germany (22), but we also received answers from other continents, like North America(19), Australia (2) or South America (1). Table 3.4 provides an overview of the participants' demographics. The job-related education of our participants can be classified as "unspecified training," "vendor training," "self taught," and "experience at the job." Most of the participants worked in a team (39), 16 were a team leader, and 10 worked alone. In the following, we report on the data gathered by the questionnaire.

3.3.3 Results

In the following, the results of the online survey are presented structured by the main categories presented in Table 3.3. The observations from the interviews suggest that company size may have an influence on different factors. To assess

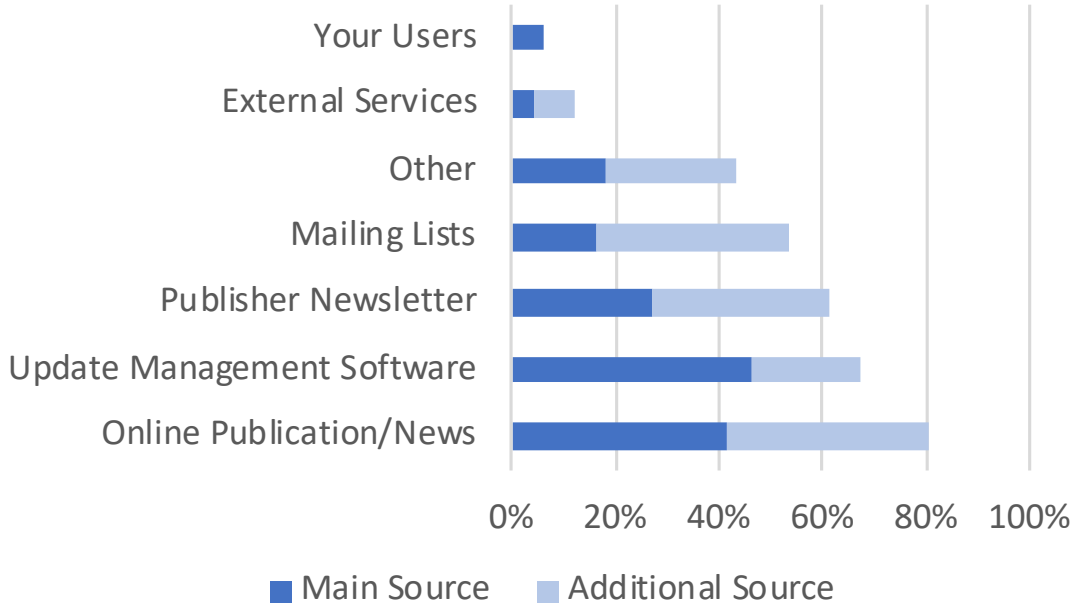


FIGURE 3.1: Distribution of information sources used by the administrators (n=67).

this point, we divided the data sets in two groups: 34 companies with 250 employees or fewer were tagged as small and medium-sized enterprises (SMEs), and 33 companies with more than 250 employees were defined as large enterprises [28]. This was found to be a suitable comparison because post-hoc we had comparable group sizes. A controlled analysis of additional factors was not feasible at this stage, and future work should consider other aspects (e.g., experience, type of systems, and team size). Table 3.5 and Table 3.6 show the answers of the participants to the statement they were presented.

Update Process and Information

U1 Figure 3.1 presents the *sources of information* administrators use to learn about (new) updates. Most of the participants reported a median of three different sources. Third-party online publications are the most frequently used sources of information. They served as a source for 54 (81%) participants, and 28 of all 67 participants (42%) even declared them the main source of information. When focusing on the main source of information, we found that update management tools are essential for most administrators (46%). Fisher's exact test indicated no statistically significant differences between differently sized companies ($p = 0.2242$). Using an optional comment field, some administrators added other sources of information, such as vendors, the online community (e.g., Twitter), work experience, and active monitoring of systems. Due to the

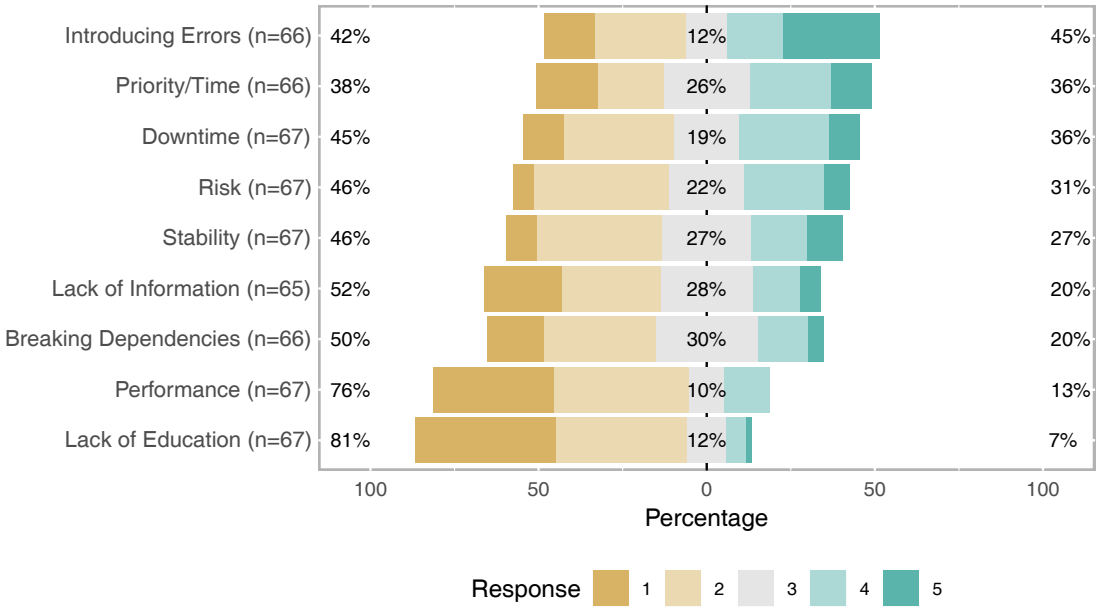


FIGURE 3.2: Frequency of considerations that hinder the installation of an update. The scale ranged from "1 – Never" to "5 – Always." Not included are "not sure" or missing answers.

structure of the questionnaire, we cannot make statements about how the participants ranked the quality of those sources. We do not know whether they use one source to get informed about the occurrence of an update and then use another to capture details.

U2 To investigate the *existence of formal update processes*, we asked the participants if 1) "there is a written document," 2) "no document but an informal guideline," or 3) "no defined process" in their company. Twenty-eight (42%) participants indicated the existence of formal processes, 26 (39%) administrators had at least informal guidelines for performing updates, and 13 (19%) participants indicated that there are no predefined processes. A comparison of the use of formal, written update processes in differently sized companies revealed a statistically significant difference between large companies (57.6%) and smaller ones (26.5%), ($p = 0.0136$, $ratio = 3.769$, Fisher's exact test). This indicates that small companies make less use of formal update processes. The lack of such a process is not uncommon in our sample, as 10 out of 34 of the small companies did not report any kind of defined process.

Update Obstacles

Figure 3.2 shows the share of administrators who have faced specific obstacles during daily update routines. Quantifying the observations, we found that general risk assessments are known to most of the participants (94%) while deciding to deploy specific updates. Only four (6%) participants answered that they never considered assessing risks as an obstacle, while 63 agreed they did so at least sometimes.

O1 to O4 When asked about more specific obstacles, or risks, *stability considerations* represented the biggest issues that had been considered by 61 (91%) participants in the past. Similarly, 59 (88%) participants considered *downtime* as a specific obstacle. *Lack of information* (50, 77%), performance issues (43, 64%) and educational aspects (39, 58%) were the least prevalent obstacles in the sample. However, even those factors were considered by a majority of the participants. Finally, we performed Mann-Whitney U tests to investigate the impact of company size on the prevalence of obstacles: We could not find statistically significant differences [9].

Fifty-five percent seemed to agree that *problems after the installation of an update* are only a minor concern. However, eight participants strongly disagreed with the statement. Five were undecided, and 25 (37%) disagreed in some way. To cover potential reasons for the answers, we assigned participants to two groups: those who do some kind of testing before installing updates on the live system ($n = 45, 67\%$) and those who do not ($n = 22, 33\%$). There was no statistically significant difference ($p = 0.2553$, Mann-Whitney U test) meaning that having a testing stage seems not to prevent all problems after the installation. Due to the sample size, we could not investigate if the company size is a significant factor in this regard.

O5 Another aspect in the interview study was the *user rights*. The agreement to the observation "Users often install software without the administrators' knowledge" was diverse. Although there was a tendency to disagree, as can be seen by the low median (3), there were also seven strong agreements. We found no statistical significance that would have supported our assumption that IT companies may have a different distribution on this than non-IT companies.

Human Factors

P1 Seventeen (25%) administrators reported that they always *feel sufficiently trained* for dealing with updates. However, 50 (74.6%) participants already faced situations for which they did not feel sufficiently trained. An evaluation of

[9] stability considerations: $p = 0.814$, downtime: $p = 0.324$, lack of information: $p = 0.655$, performance issues: $p = 0.067$, educational aspects: $p = 0.752$, introducing errors: $p = 0.611$, risk considerations: $p = 0.415$, breaking dependencies: $p = 0.387$, priority: $p = 0.559$

Interval	Number
Hours to a day	11
Within a week	19
Within two weeks	8
Within one month	11
More than a month	9
No answer/no usable information (e.g., missing unit)	11

TABLE 3.7: Reported time intervals between the release of an update and deployment on all systems.

the impact of the administrator's company size indicates that administrators at large companies ($Median = 4$) more often feel sufficiently trained than their colleagues at smaller companies ($Median = 4$), Mann-Whitney U test: $U = 358.0$, $p < 0.01$, two-sided.

P2 Finally, all administrators except one somewhat agreed that *timely updates are important*. The self-reported time span between the release of an update and its installation can be seen in Table 3.7. While some participants reported deploying updates within a day, there were nine cases where updates needed more than a month. Optional comments given by the participants supported the findings that downtime, complexity, and dependencies are common reasons for such delays.

(Missing) Distinction between Security- and Feature-Updates

The interviews revealed that security- and feature-updates are often hard to distinguish. While we did not ask for the share of security-related patches in our interviews, the survey participants reported that 56% (ranging from 5-100%) of the overall updates involved security-related ones.

3.4 Discussion and Implications

Our work identified multidimensional problems that should be addressed by multiple stakeholders (e.g., software vendors or the companies themselves). In this section, we reflect on our results, provide actionable recommendations for these stakeholders and suggest directions for future research. We acknowledge that many aspects reported in this paper may seem like "common sense". With this work, we add to the scientific evidence in this very broad area with several factors that influence the update process and directions for further research and discussion.

3.4.1 Security Implications

Our results are in line with Li et al. and show that even professionals cannot always deploy updates in a timely fashion. This can be a security issue since outdated systems are often vulnerable to exploits. The administrators we asked were aware of this problem and agreed that deploying updates in a timely manner is important. However, we found that external factors such as compliance with company-specific rules, inflexible processes and communication overhead (e.g., leadership approval) still delay updating in practice. Future work needs to take a more holistic view and investigate technical and social factors in the update process. We need to understand which people are involved in these processes and how their communication can be supported. In addition, we need to develop approaches to better communicate the urgency of specific patches as today, the rating is often not clear.

3.4.2 Update Process

The results showed that the update processes of system administrators are diverse and complex. Although the update processes of administrators can be matched to the end user stages [131], the identified stages differ in the details. In particular, gathering information and discussing update decisions were identified as important but time-consuming steps. As many administrators reported they make decisions in group meetings, we raise the question of how individual administrators can be supported in their decision-making process. The preparation process takes time and involves extensive testing. Although the testing processes were handled differently, they usually involved multiple iterative stages. This indicates that administrators have to go through the whole update process multiple times. Two findings were primarily interesting: 1) Many companies lack formal processes, and 2) the update process is highly complex and lacks automation. The insights into this process provide important directions for future research and immediate action items for software vendors, such as the following:

- Formal processes seem to be more frequently used in large companies. Whether formal processes help to reduce the burden of decision-making and ease the overall process should be researched; that is, in what way they influence the update process (e.g., can well-defined responsibilities speed up the decision and do they lead to more and faster updates?) and where possible trade-offs can be expected (e.g., decreased complexity versus more time needed).

- The high number of iterative steps must be supported, e.g., with automation approaches. Thus, it is important to understand which stages of the process are critical and which parts can be effectively supported by tools.

- A possible approach for improving the process could be to connect more effectively virtual teams of administrators who share similar responsibilities and manage similar systems. Supporting such concepts with feasible tools can quickly lead to shared knowledge of best practices and experiences resulting in a better overview of the effects updates have on their systems. We hypothesize that especially smaller companies would profit from that.

3.4.3 Obstacles

The findings indicate that administrators face severe obstacles that often hinder them from performing timely updates. In line with Dietrich et al.'s work [32], the findings show that the problems administrators face are diverse and interconnected. Corresponding to Hrebec and Stiber's findings [70], individual-related factors, such as negative and positive experiences with updating, as well as education, come into play. The findings provide a baseline for future research questions and immediate action items for software vendors, such as the following:

- Due to the highly diverse landscape of large-scale systems, future research should further explore contextual factors and different populations of administrators. Differentiation of the various types of administrators could help to better categorize participants and understand their diverse problems and challenges. Related to this point, the check of the external validity of the research would benefit from better differentiation of types of administrators. However, a practicable taxonomy for this is still missing.
- Software development should focus on reducing downtime and providing rollback mechanisms that encourage administrators to take the risk of potential negative effects on availability.
- Researchers and software vendors should investigate on how to provide reliable information and accurate documentation of the effects of an update and occurring problems right in the moment and at the place the update is going to be installed.

Therefore, we hypothesize that supporting administrators' situational overview will have positive effects on timely updates. Finally, minimizing consequences by providing reversible updates, or just updates that have very small effects, could furthermore help administrators to update. As an example, dynamic software updates (DSU) [65] seems like a promising technique to contribute to this area and could be evaluated from this perspective.

3.4.4 Coping Strategies

As a consequence of facing obstacles, system administrators have developed a diverse set of coping strategies. Although the degree of usage varied among

participants, an important countermeasure against the growing complexity is the use of tools that monitor update processes and support to (partly) automate installations. Because administrators expressed the desire for more automation, the findings emphasize the importance of the area of research that deals with the development of such concepts [16, 56, 81, 100].

To cope with the problem of limited resources combined with growing package sizes, the participants started to divide update processes into multiple batches. This can have the advantage of allowing more feedback loops and of reducing the load on the network. However, at the same time, this process increases the number of required iterations for single patches. Although we argue that the footprint (e.g., resources needed to roll out), especially of security updates, should be minimal, this may not always be possible.

Based on the findings, we provide the following recommendations to support existing coping strategies and for the development of novel solutions:

- Hot swap functionality and small-sized patches which enable administrators to estimate the impact of the installation on their systems, have the potential to further ease the update processes.
- Update management tools should better support the integration of third-party software.
- Administrators' coping strategies are still not sufficiently understood. Thus, researchers should focus on systematically investigating different coping strategies for various obstacles, identify desirable behavior and analyze in which way the human aspect contributes to this.

3.4.5 Comparison to Results by Li et al.

As mentioned before, a thematically similar publication emerged independently while we were working on this research. Li et al. published a study on system update processes among US American system administrators, identifying an update process that was very similar to ours [84].

The Update Process

While Li et al.'s process emerged entirely from their interview response data, our update process was informed by theoretical work by Vaniea et al. [131]. This could explain minor differences such as the separate testing stage we introduced to highlight the difference to the end user process.They found that admins go through five stages when updating. First, they become aware of a new update (*learning about updates*). Second, they need to decide whether or not to deploy it (*deciding to update*). In the next stage, the preparation for an update is done, e.g., making backups or preparing machines (*preparing for update installation*). Following this stage, there is the deployment itself, including coordination of

when to update (*deploying updates*). Finally, post-deployment issues are handled (*handling post-deployment issues*).

While both update process models are very similar, they also show differences when looking at them in detail. In Figure 3.3, an overview of both models can be seen. In case a stage includes the same tasks in both models, only the name of the stage is given (e.g., 1. *Learning about Updates / Information*). However, if a certain task was mentioned in different stages, the task itself is explicitly mentioned and color-coded. While Li et al. [84] include the task of "testing an update" in their third stage, we awarded testing its own stage. Additionally, we mentioned non-technical preparations as coordination in their preparation stage. Li et al. [84], however, include this step in the deployment stage itself.

Going through the stages in sequence, in alignment with Li et al.'s findings, we can confirm that in the information-stage, administrators use multiple sources to derive information about updates. We didn't find any statistical difference in the number of sources used between administrators working in different companies (big vs small) in our sample. Li et al. reports on the frequency of the used sources and that three quarter of their participants used security advisories or direct vendor notifications. In our data, 81% informed themselves using online publications and 63% relied on publisher newsletters. We can add that despite having multiple sources (median=3), our population uses update management tools as their main source followed by online resources.

Both works identified the deciding-stage. We can match most of our identified obstacles to the reported factors of Li et al. With a slightly different perspective, we can add an additional reported obstacle that focuses more on the administrator executing the process than the update: missing expertise.

We can support Li et al.'s finding that testing is an important stage in the process and we encountered the same approaches: "Staggered deployments" and "Dedicated testing environments". As 83 of 102 (81%) of their survey participants included some form of testing, a slightly smaller, but still the major, part of our participants 45/67 (67%) reported the same.

As for the remaining two stages, our works differed in focus. While Li et al. extensively discussed the method of deployment (automatic vs. manual) and the decision of when to deploy in the deployment stage, our work concentrates on the obstacles the administrators face in this stage. For the post-installation stage, their work presents the ways in which administrators deal with update issues, while we report on the frequency of the occurrence of such issues (O3) in section 3.3.3.

Obstacles in the Update Process

Li et al. identified challenges faced by administrators within this update process that can be categorized as: (1) obtaining relevant information about relevant updates and deciding, (2) preparing, testing and deploying updates in a timely fashion, (3) recovering from update-induced errors, and (4) organizational and

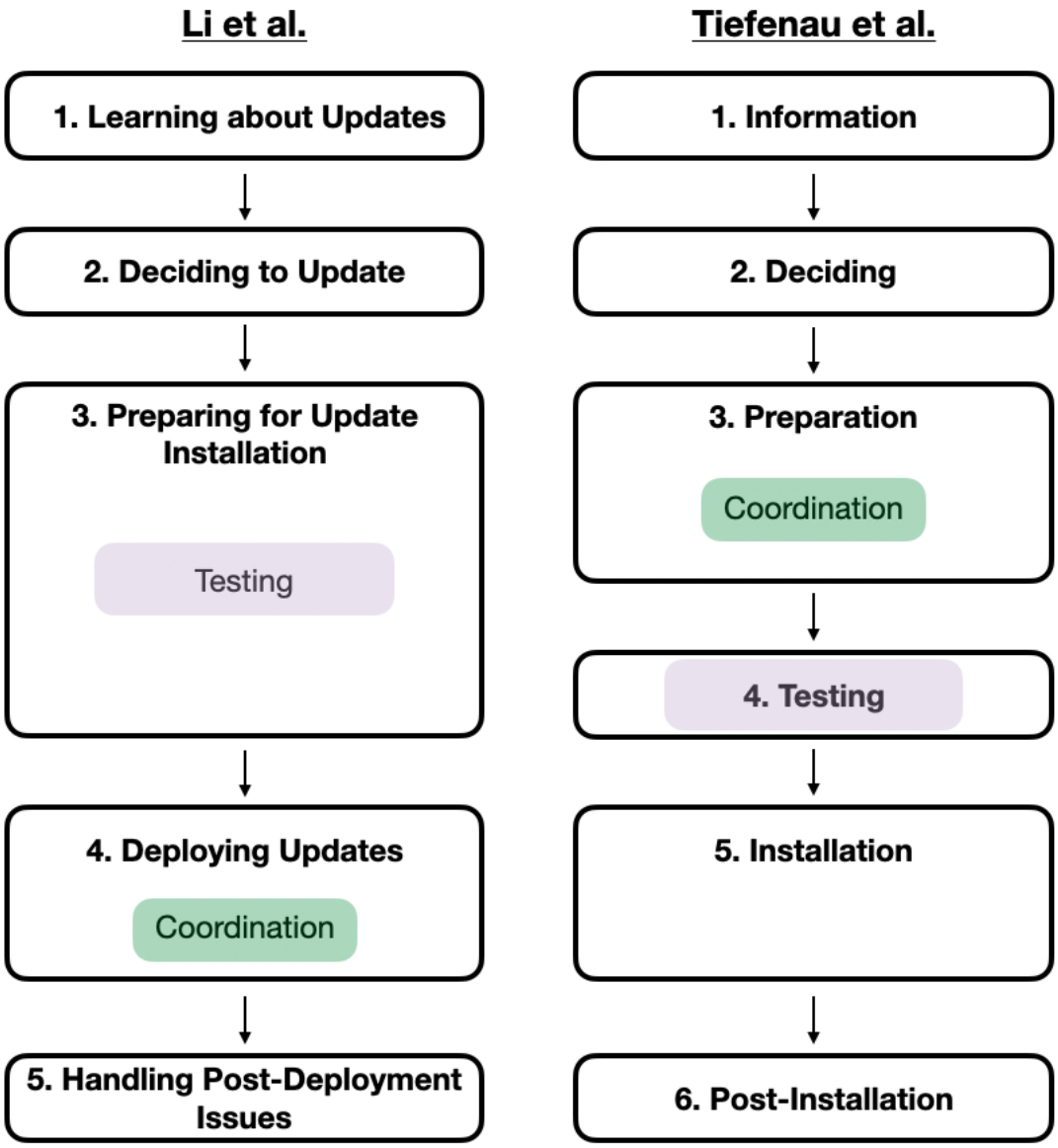


FIGURE 3.3: Differences in the update process model of Li et al. [84] (left) and ours (right). Only the differences are color-coded.

management influence [84]. Our identified obstacles (cf. section 3.2.4) are in line with these obstacles. Li et al.'s work reports that identifying the relevant information in an update can be a challenging task. We can confirm this (O4) and show that this was mentioned by 77% of our participants.

Automation can help to deploy updates sooner and more frequently. Li et al. have found several obstacles such as dependency and compatibility considerations or host heterogeneity as factors that have an influence on update deployment. In addition to those, we have found additional ones such as missing tools or performance considerations in our data set. Table 3.2 provides a summary of our findings that assigns the problems to the stages in which they occur.

In general, while their work reveals the existence of those problems, we can complement these problems with the frequency of the problems that our survey participants stated. Li et al. report that the recovery of updated-induced errors is a problem that we can enrich with the fact that this seems to be of mixed importance (O3). This could indicate that this is a context-dependent factor, and a more detailed research must be undertaken in this regard.

Also, Li et al.'s work reports on the existence of organizational oversight that hinders or delays updates in some cases. We can also find this problem and show that this, among stability and risk considerations, is of more importance than factors such as performance considerations.

Demographics

While both Li et al.'s and our study are very similar in methodology, they differ in a key point: the recruited sample. Li et al. sampled only US-based administrators, while we recruited our interview-study population from Germany and our survey participants were mostly (41 of 67) European-based. Despite work culture in the US and Europe (e.g. in Germany [64, 104, 47]) being distinctively different (stemming from cultural differences in education, law, and professional socialization, among others), both studies report similar findings. We are thus in the fortunate situation to not only have our methodology and findings independently validated within a close distance in time, but also to confirm that the phenomena we identified are relevant across both US and European system administrators.

On interpreting the independently compiled findings, we have an indication that the system administration process is not as susceptible to cultural differences (at least in Western societies) as other fields of work. This might be connected to the rather globalized nature of IT infrastructure. Both participant pools used similar software, e.g., SCCM or WSUS (cf. section 3.2.4). It is reasonable to assume that the technical challenges are similar. Comparing both papers, we could not find any differences that originate in individual or organizational factors. If this can be confirmed in further studies within different countries such

as China (the largest producer of IT hardware and systems[10]), Estonia (the often considered "most advanced" country within the EU in terms of digital transformation[11]), or Qatar (the largest economy in the Middle East according to GDP per capita [12]), this would significantly widen the recruitment possibilities for future studies within the field of system administration.

3.5 Limitations

The population we refer to as administrators is inherently diverse in terms of responsibilities, education, and previous experience. Depending on the size of a company, administrators have different responsibilities and work either in isolation or in larger teams. Furthermore, the security requirements depend on the types of products and services a company offers. Also, there is no unified career path for administrators, and one must not necessarily have a degree or certificate of any kind to become an administrator. Because of all these aspects, the results are not generalizable and thus applicable other populations of administrators with different demographics or training. The participants in the online survey were mainly from Europe and the United States. In these regions, technical staff like administrators are predominantly male which is why the sample was heavily biased in terms of gender. Due to our recruitment strategy for the quantitative study, the sample potentially suffered from self-selection bias, as was likely also due to the completion rate (51.1%) of the survey. Regarding our questionnaire, we did not ask the participants about their current employment status. This could result in answers from people that worked as an administrator previously and are now in a different position. However, due to the mentioned self-selection bias we think that the participants are still somehow active in this area. Also, we did not collect information about the systems and software, the administrators were in charge of. Because of this, we cannot report possible existing differences between, e.g., different operating systems or widespread versus niche software. The analysis is based on self-reported data, and thus, participant reports are highly subjective. We have no reason to believe that social desirability and recall bias are uncommonly strong in the sample because the interviews and related work showed that administrators tend to admit that they do not know about everything [70]. However, this must be taken it into account, especially when talking about risk, obstacle perception, and individual perception (e.g., P1). Finally, the qualitative interviews provided useful insights but did not reach saturation (cf. [53]). However, the potential lack of saturation

[10] https://www.mckinsey.com/~/media/mckinsey/featured%20insights/china/china%20and%20the%20world%20inside%20the%20dynamics%20of%20a%20changing%20relationship/mgi-china-and-the-world-full-report-june-2019-vf.ashx

[11] https://www.wired.co.uk/article/estonia-e-resident, accessed 11/21/2019.

[12] https://www.cia.gov/library/publications/the-world-factbook/rankorder/2004rank.html, accessed 11/21/2019.

is alleviated as the qualitative analysis was primarily used as an exploratory first step to build hypotheses. The answers to the free-text questions on the questionnaire did not bring up many new topics which make us confident that the most common real-world problems were covered. But, although several different issues were covered, we make no claim for completeness.

3.6 Ethical Considerations

At the time this study was conducted, the computer science department of the University of Bonn did not have a formal IRB process for this type of study but has a series of guidelines to follow. According to these guidelines, we limited the collection of personal information as much as possible and collected data separately from contact information. Furthermore, all the processes complied with the European General Data Protection Regulation (GDPR). As the administration of services in a corporate environment is a sensitive topic, we did not collect detailed information about the companies' infrastructures. In addition, participants were explicitly given the chance to drop out at any time during the study. Finally, we emphasized the option to skip questions that participants preferred not to answer.

3.7 Summary

This chapter contributes a mixed-methods study that revealed how administrators incorporate security updates in their daily work routines, what obstacles they experience, and their coping strategies. We found that even experienced administrators find it hard to predict the direct consequences of applying an update and are heavily concerned about potential downtimes. Another interesting observation was that administrators often rely on information not provided by the (software) vendor but by online media or by their peers, who often face similar struggles. Among other things, the findings imply that there are aspects that vendors can influence, such as providing sufficient documentation or more granular updates, which can help to motivate administrators to update and support them in the update process. This fact is revisited in chapter 5.

Early on in the interviews, we found indicators that other stakeholders influence on the update process. We had the chance to work together with a company to observe this fact and also had the opportunity to apply our created model. I present the results of this study in the next chapter.

Chapter 4

A Case Study on the Update Processes in a Corporate Context

Disclaimer

At the time of this work, this chapter's contents are under review as part of the paper "One Process does not fit All: A Case Study on the Update Processes in a Corporate Context" at the USENIX Security conference 2021. This was joined work together with my co-authors Maximilian Häring, Eva Gerlitz, and Matthew Smith. As this work was also conducted with my co-authors as a team, this chapter will also use the academic "we" to mirror this fact. This study was part of a master thesis done by Ronald Brenner, who was also working in the observed company and gathered the data. The idea and initial concept for this work came from myself and Maximilian Häring. Ronald Brenner conducted the interviews and the survey. Maximilian Häring and I coded the tickets and, together with Eva Gerlitz and Matthew Smith, generated the new proposed model for that I prepared the results by analyzing the dataset. Before compiling the paper for publication, Maximilian Häring, Eva Gerlitz, and I jointly discussed the study's implications.

4.1 Motivation

To validate the model of chapter 3 and to further investigate the influence of different stakeholders on the update process, we conducted a case study in a German web development company that managed web content management systems (WCMS) for their customers. We used an ethnographic approach by having a researcher working in the company. Also, we analyzed 116 update related processes extracted from their ticket system. Coding these tickets using the stages of related work revealed that the update processes we observed did not map to those of Li et al. [84] and the previously proposed model. This happened as the stages alternated and reoccurred within the data set. In this chapter, we build and present an extended model of the update process in corporate environments that is presented in section 4.4. This model combines both

existing models and captures the update process in a more flexible way by allowing back-and-forth transitions between the stages. Also, it takes external factors, e.g., getting aware of a further update, into account. We state that this allows representing a larger number of processes in very diverse contexts. The rest of this chapter is structured as follows: In section 4.2, we present the studied company and the methodology. In section 4.3, we show the results of the coding process and where the models are not flexible enough, and in section 4.4, we present the extended model.

4.2 Methodology

We conducted a case study to observe the update process in a German web development company (in the following called DevComp) by analyzing tickets from their internal ticket system. Before doing so, in 2017 and 2018, a researcher working at the company conducted interviews and a small survey to gain deeper information about the participants and the company itself. The protocols can be seen in section B.1 and section B.2.

The following section first presents relevant information about the company and the participants that was acquired through the interview and survey. This is followed by the methodology regarding the ticket system.

4.2.1 Company

First, we explored the given infrastructure of DevComp by conducting interviews with all employees except the two Co-CEOs. With this, we aimed at finding answers to the following questions:

- What employees are involved in the update processes?
- What update workflows exist?
- What roles and responsibilities are defined within an update process?
- What software exists that needs getting updates?
- Which workflows are suitable for a further examination?

Company Structure

By the time we conducted the study in 2017 and 2018, DevComp held four different departments. Figure 4.1 gives an overview of the organizational structure, including the number of people involved in the updating processes. The top-level is the management, consisting of two Co-CEOs. The remaining compartments all belong to the second level:

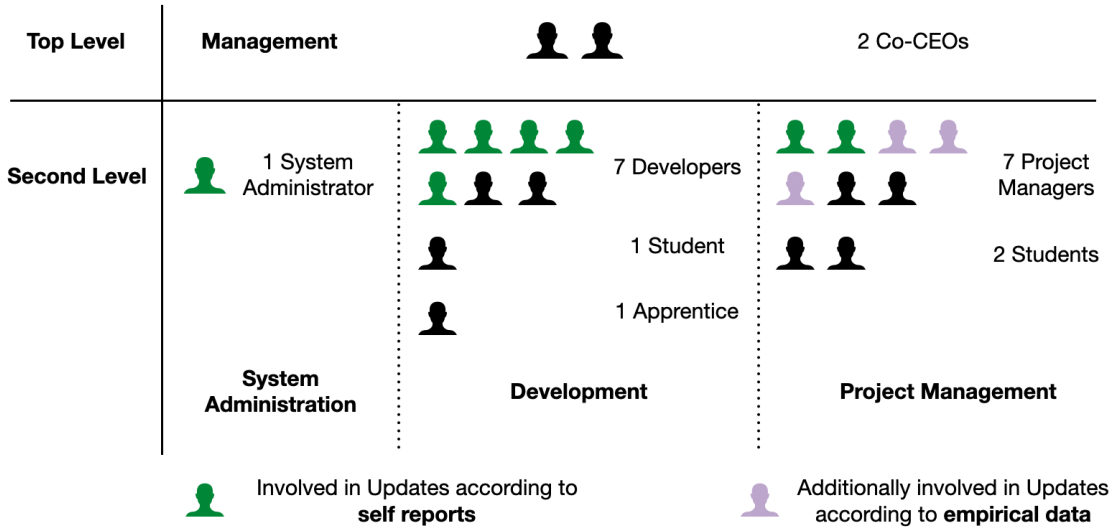


FIGURE 4.1: Structure of company. Green indicates involvement in updates according to self-reports. Purple indicates those persons that did not mention to be involved in updates during the interview but later showed up in update tickets. Black neither mentioned to be involved in updates nor turned up in tickets.

- System Administration (consisting of one system administrator)
- Development Compartment (consisting of seven developers, one working student, and one apprentice)
- Project Management (consisting of seven project managers and two working students)

Based on the self-reported data (interviews and surveys), eight employees indicated to be involved in update processes (colored in green). However, we identified three additional project managers who also worked on tickets concerning updates when we looked at the tickets (colored in purple). We will further look into this in the discussion (section 4.4). All departments in DevComp work closely together, and the flat hierarchies allow short communication channels. Most project- and task-oriented communication is handled via a ticket system that includes both, communication within DevComp, and with customers, who get limited access to the ticket system. The participants reported that within their company, some of the communication happens outside of the ticket system via face-to-face, mail, or phone, which we could confirm in the analysis of the tickets ("As mentioned on the phone...").

Updated Software

Based on the interviews, we were able to identify three different types of technologies that received updates:

First, server updates that include all patches to server software such as PHP, Apache, or MySQL. All of these are required to host high-level applications. Most of these applications run in virtual machines that are hosted on servers of an external company, but there are self-hosted internal servers as well. The system administrator updates the whole infrastructure, and the installation usually happens without further communication if it does not imply unplanned downtime. To allow this, an agreement was made about the time updates can be deployed in general, without interrupting the staff during working hours.

Second, we found high-level applications like WCMS, analytics, or newsletter applications hosted on behalf of the customers. These are maintained and updated by project managers and developers. Usually, the project manager and customer schedule an update, test update effects and share the feedback with the responsible developer.

Third, custom applications and used libraries need to be updated, which is done by the responsible developer. Library updates are autonomously planned and executed by the developers. Since they are mostly deployed with other, already planned updates, no particular customer arrangement is needed.

Shared responsibilities: The update-performing employees were asked for their responsibility for projects and with whom they share it. A list of these responsibilities can be seen in section B.2. Both project managers, who indicated to be involved in updates during the interviews, only hold shared responsibilities. Updates are always delegated to a developer[1]. The developers maintain updates in cooperation with at least one coworker, most often with one project manager, and in a few cases, other persons were consulted. In contrast to that, the system administrator mostly works independently: Only one of his 13 mentioned responsibilities is shared with a project manager.

Based on the interviews, the ticket system, which is used for internal communication, is more often used by project managers or developers than by the system administrator, who usually works independently. On the rare occasion that server updates that are executed by him also needed further communication, the discussion also happens within an issue. While it seems to be the communication tool when handling updates, neither of the employees mentioned it as a helpful tool explicitly for simplifying updates. Regarding what they deem helpful in the update process, project managers and developers mentioned installation tools like Composer or npm. The system administrator, who mentioned to use the ticket system seldom, mentioned tools like Ansible.

[1] At least this was the case in the interview phase. Later, we learned that in some projects with easy installation-processes of updates, e.g., by just clicking the button “Update now”, the project manager tries to do this task before consulting the developer.

Ticket Type	Count
Update ticket (single)	116
Update ticket (multiple)	38
PHP7 Upgrade	58
Post-installation problems	21
Incomplete	24
No information	13
Post-installation task	5
Not update-related	18
Total	**295**

TABLE 4.1: Ticket types and the number of tickets assigned to each group.

4.2.2 Ticket Analysis

After conducting the interviews and surveys and learning about the company, we started with a set of 31327 tickets from their ticket system between 2008 and 2018. In this database, we filtered 295 issues that included the word "update" or "upgrade" in their description or notes. Afterwards, we looked at this list of tickets. We manually checked them for relevance concerning the update process and further information, such as the software that needed to be updated. For further analysis, we assigned them to certain ticket types. An overview can be seen in Table 4.1.

Within the ten years, the company faced a massive update from PHP 5.4 to PHP 7 for a project. The company handled this update by breaking it down into small pieces and creating many tickets for this purpose. As this would distort the analysis, we excluded these 58 tickets from the analysis. We further excluded 38 tickets that included more than one software that needed an update[2], 26 tickets that only handled post-update problems or tasks without information about the already deployed update, 24 that were not finished and 13 that were not about the task of updating itself but, e.g., a ticket collecting references to others. At last, 18 tickets were excluded, because they were not software-update related. In them, the word "update" was, for example, used as "update on project X". The resulting set of tickets that we identified describing exactly one update process had a size of 116 tickets.

In total, we could identify 24 projects that consisted of one (usually WordPress) to six software products. WordPress was the most common one and appeared in 17 projects, followed by Joomla, which was used in ten and Piwik/Matomo (8). Eight software products (e.g., Perl, Limesurvey, HA-Proxy) were used only in one project and appear as "Other".

[2]Commonly, this was a ticket that contained the task to update to a specific version, but for multiple projects that used the same software.

Software	Days opened: Min	Max	To installation Mean (sd)	To closure Mean (sd)	Projects	# of tickets
WordPress	1	189	16.1 (29.2)	17.3 (30.1)	17	57
Piwik/Matomo	2	75	11.6 (17.9)	15.3 (17.3)	8	16
Joomla	0	140	27.8 (44.7)	42 (41.5)	10	14
WordPress-Plugins	0	39	7.9 (9.75)	8.7 (11.7)	5	10
TYPO3	7	140	59 (48.4)	60.3 (49.6)	5	6
Imperia	65	144	94.2 (24.3)	98.6 (30.6)	4	5
Other	0	217	48.6 (59.8)	69.8 (80.3)	7	8

TABLE 4.2: Total number of update tickets and time to installation and to closure (in days) for each software. "Other" includes software that only appeared once. Projects denotes the number of projects (out of 24) in which the software was used.

Further details about the software in the analyzed tickets can be seen in Table 4.2. It shows the number of update tickets grouped by software and the number of projects in that they occur. WordPress is the most frequently updated software, followed by Piwik/Matomo and Joomla. Despite TYPO3 having released 40 updates in the same period, which is nearly the same as the number of WordPress updates (47) in the same period, the software only appears in six tickets. We could not find information about why the company skipped most of the smaller updates but found the motivation to update in one case: due to the end of support for a long-term support version (7) in 2018, they decided to update to a new version. TYPO3 had no CVEs with a score of 7 or higher [27].

The tickets contained general information like the internal ID, the current responsible person to fulfill the task, and the date of creation, but also a field for a brief summary and a description. In the description, the employees usually wrote the software and version that needed an update and sometimes information about the update itself. Following that, there is a timeline that contains notes in which the employees can write messages to inform their colleagues about the steps or decisions they have made. So, in the end, each ticket consisted of one or more notes from the staff members who worked on the ticket. Each note had a date, a person who was responsible, and information text. We coded the stages those notes belonged to. A note was coded based on the note description that contained information about what the author had done and what had happened up to this point. An example for this can seen in Figure 4.2.

The coding process looked as follows: In multiple steps, two researchers tried coding the whole set of 116 tickets following the stages, as proposed in Li et al. [84] and in the previous chapter. In a discussion, it was agreed that this is not ideal to describe the process within the ticket system. The previous models assume a stricter order of steps, as described in subsection 3.4.5 of chapter 3. The coders added further codes to cope with differences in the models and allowed an arbitrary use of the codes in order to remove the restrictions. The used codes

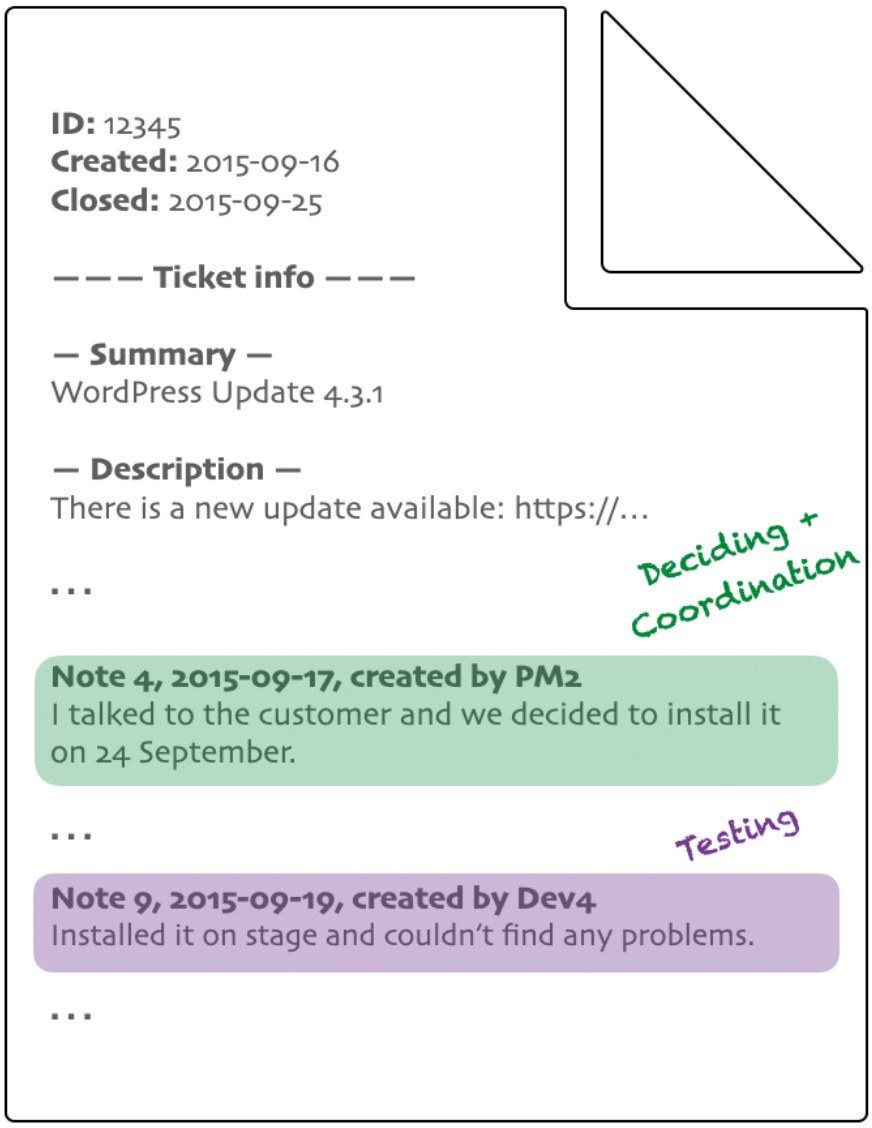


FIGURE 4.2: Example of an excerpt of a coded ticket. The codes indicate what had happened previous to the note.

are described in detail in section 4.3. The same two researchers coded ten tickets using the new codebook. The calculated Brennan and Prediger inter-coder agreement was 0.80 [23]. Following that, each researcher went through half of the tickets and coded them again.

4.3 Results

In this section, we present the results for the tickets with a focus on the coded stages. In the following, when we quote examples from within the tickets, we indicate a ticket pseudonym, the number of the note within this ticket, and the author's role. As all tickets were in German, we translated them into English.

4.3.1 Stages/Codebook

Learning We coded notes in the *learning* stage, where the author reports that they became aware of a new update somehow. This is similar to the *learning about updates*-stage of Li et al. [84] and to the *information*-stage of our first study in chapter 3 (in the following indicated as *learning about update/information*). In all except three tickets, this was given implicitly because at the point the ticket was created this step was already finished. However, we found six tickets in which a person reported to have found a new update within the update process itself: "Today, a new Piwik update was released. Does it make sense to deploy this directly?" [Ticket A, Note 9, Developer].

Deciding Notes were marked as *deciding* (*deciding to update*[84]/*deciding*), where decision processes were mentioned. For example, when the company was negotiating with the customer about the costs of an update and the impact on the systems, the customer said: "Thank you for the offer. This is OK. However, we have to assure that we do not exceed the [planned] hours for this update." [Ticket B, Note 4, Customer] or "Fine. Please install it." [Ticket C, Note 4, Customer]

Preparation & Testing Related work differed in the preparation code: In the first study of this work, we proposed *testing* to be a stage, while Li et al. grouped it into the preparation stage. We coded *testing* separately because it allows us to group testing and preparation to resemble the model of Li et al. [84]. We labeled those notes as *testing* that contained information about the testing process itself, such as "The update is deployed on the test system" [Ticket F, Note 6, Developer], but also the results ("it works wonderfully" [Ticket G, Note 3, Project manager]) and fixes in the process ("... there were missing permissions, that I have now granted" [Ticket H, Note 9, Developer]).

Notes were coded as *preparation* (-/*preparation*) if they contained topics that are related to non-technical preparations for testing, like internal task assignments or agreements. Furthermore, preparing the technical requirements that are needed for testing also fell into this category: "Can you take on the realization of a workaround?" [Ticket D, Note 11, Project manager] or requesting data for the technical requirements for building a staging system [Ticket E, Note 15, Project manager].

Deployment & Coordination Following the testing stage, we separated the *deployment* stage from Li et al. [84] into two codes, as there were differences to our (*installation*) in chapter 3: As *deployment*, we coded those notes for that we were certain that the update was deployed on the live system: "The update was deployed on the live-system" [Ticket I, Note 7, Developer]. We also introduced *coordination* that was created for information that was not the direct technical deployment, but rather involved steps to prepare the installation on the live system. For example, this included agreeing on an installation date or who is going to deploy the update. In the model created in the previous chapter, this step was included in the *preparation* stage, Li et. al. [84] included it in the *deployment* stage, as shown in Figure 3.3.

Post-Deployment Each note that came following the successful deployment of an update was coded as *post-deployment* (*handling post-deployment issues[84]/post-installation*). Here, communication with the customer, as well as troubleshooting after the installation and closing remarks, happened. Unsuccessful deployment was not coded as *post-deployment*, as the model of Li et al. [84] suggests. This is relevant for tickets where an installation failed. Sometimes, backups were rolled out, and after searching for a solution and coordination of a new installation date, the installation was successfully done.

Stage Transitions

After coding each note, we analyzed the flow in every ticket. With flow, we mean the appearance of codes in the tickets ordered by the note numbers. They do not really match the intuition of progress of a linear path, as we observed many back-and-forth transitions between nearly all stages. The amount of the transitions between each of the stages can be seen in Figure 4.3. Figure 4.4 gives a graphical summary of all observed transitions with those that did not occur in related work marked in red. It suggests that the update process, especially before the deployment, is not as linear as suggested in the models of Li et al. [84] and the work in chapter 3. The Figure does not take into account when the notes received more than one stage. For example, one note included information about a newly released update that deprecated the update initially discussed in the ticket. On this very same note, the decision to deploy the new update was made as well: "A new security update was released last night once again. Could you please deploy it?"[Ticket J, Note 4, Developer]. Due to the methodology, we cannot make statements about the order in which steps are done between notes. Hence, the numbers have to be seen as an upper bound for the case study. In the following, we present some examples of the transitions between stages:

FIGURE 4.3: Heatmap of stage transitions in the data set based the update process model of Li et al. [84] (upper) and our previous model (lower). The black framing indicates transitions that are expected using the models. This includes either staying in the same stage (*Deciding -> Deciding*), or a transition to the following stage (*Deciding -> Preparation*).

Link back to Learning

In *preparation*, *testing* and *deciding*, we could find at least one ticket where the process switched back to the *learning* stage. In these tickets, one of the involved persons found a new update for the software currently discussed in the ticket. Therefore, the process may need to go through the learning and decision stage again. It could be argued that in this case, this should be modeled by another update process. However, it influenced the ongoing process; for example, if it was decided only to deploy the newest update, testing did not proceed for the original one.

Link between Deciding and Preparation

Very often, the *preparation* stage followed the *deciding* stage. However, we also found examples for the other way round. This occurred, e.g., because the *deciding* stage is not solely defined as the simple decision to install, but also to clarify the financial situation. For example, a ticket starts with estimating the time needed and the internal assignment (*preparation*). This information is then passed to the customer, who decides whether to install it or not (*decision*).

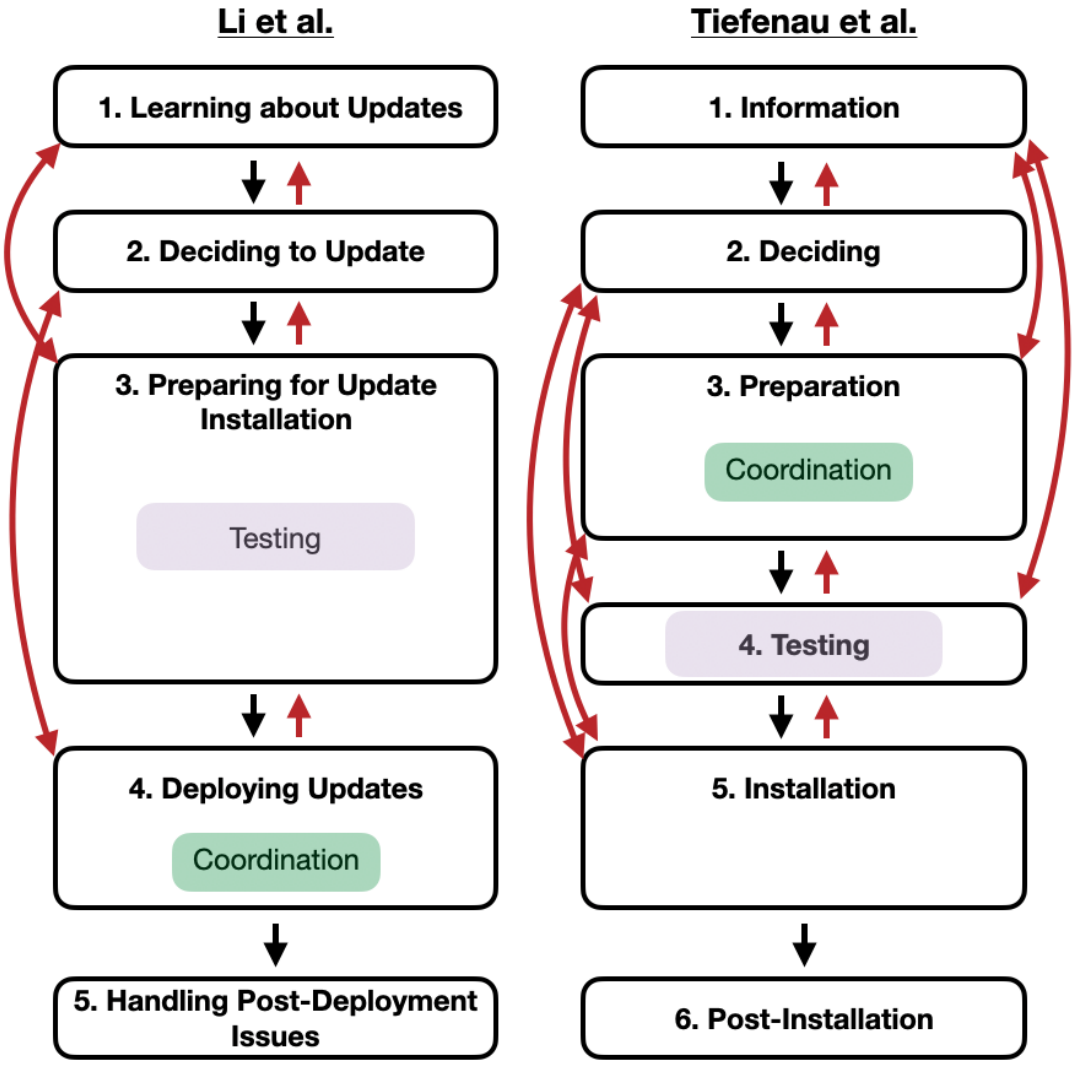


FIGURE 4.4: Observed stage transitions in the data set when mapped onto the update process model of Li et al. [84] (left) and our previous model (right). The red arrows indicate new transitions that are not mentioned in related work.

Link between Deciding and Testing

When separating the *testing* stage, as proposed by in the first study, we found one jump from *testing* to *deciding*. In this case, a new update was released during the testing stage and the decision to install it directly was done in the next note. However, the transition from *deciding* to *testing* was the more common case.

# of Persons involved	Days opened			Ticket
	Mean	Sd	Median	Count
1	18.3	30.9	1	3
2	10.3	17.3	5	49
3	23.2	27.5	14	39
4	63.8	51.7	49	12
5	71.6	59.6	39	9
6	45.5	3.54	46	2
7	217	-	217	1
8	140	-	140	1
Total	28.6	41.8	9	116

TABLE 4.3: Ticket times (in days) based on the number of involved persons in the update process.

Link between Preparation and Testing

We found transitions in both directions between *preparation* and *testing*. A common theme from *preparation* to *testing* was the gathering of information for testing before installing the update on the staging system. The other way around, an example was a project that required a backup which we coded as *preparation*.

Links between Deciding/Preparation/Testing and Deployment

Using Li et al.'s [84] model, we identified transitions from *deployment* to the *preparing/testing* stage. Following Li et al. [84], tasks, such as timing the update or the internal coordination, belong to the *deployment* stage. We coded them as *coordination*. The same reason is responsible for the transition between *deciding* and *deployment*. We saw coordination tasks frequently occurring at the beginning of the process or interwoven with the testing process.

Some tickets also skipped the *testing* stage altogether, resulting in the direct connection between *preparation* and *deployment*. We could also observe some instances in which the first note was the deployment itself. In those, preparation and testing were not present in the system (but certainly happened).

Deciding not to update

We observed eight tickets in which there was no deployment of the update. In those, decisions were made in the progress that resulted in not installing the software. For example, after talking to the customer, they agreed that "the update is not necessary anymore, because [the customer] will stop working with Piwik in a few weeks" [Ticket K, Note 4, Customer].

4.3.2 Involved stakeholders

We looked at the number and role of involved persons in a ticket. In most of the tickets (88 of 116), two or three persons appeared in the process. Most of the time (n=43), this included a project manager and one developer. The second frequent combination was a project manager with a customer and either the administrator (n=14) or a developer (n=7). Table 4.3 shows the mean and median time of the tickets grouped by the number of involved persons.

As mentioned before, we learned that project managers try to install an update when possible. This, although anecdotal, is evidence that even in a professional setting, the update tasks themselves are shared and sometimes executed by untrained management people.

4.4 Discussion

By applying real-world data to existing update process models, we identified that the models were not a good fit for the collected case study data. We, therefore, propose a model that adds flexibility to the order of the stages in the process.

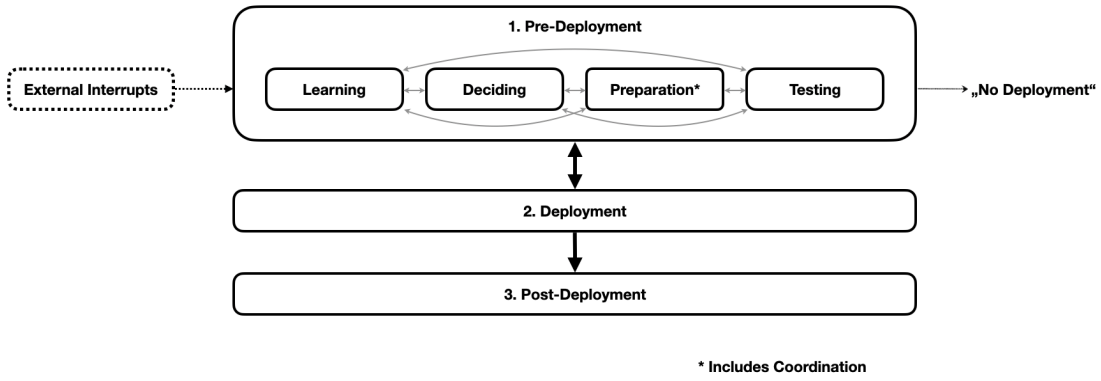


FIGURE 4.5: Adapted model to describe the update process in a corporate context.

Figure 4.5 shows a visual representation of the adapted model. We observed that certain stages are not fixed in a specific order. While we found tickets that followed the straightforward model of previous work, many tickets showed jumps, as demonstrated in Figure 4.4. This seems more understandable when one understands stages not as steps that have to follow each other but more as a grouping of actions that somehow relate to each other by having a common goal.

We therefore grouped the stages *learning*, *deciding*, *preparation* and *testing* into a *pre-deployment* stage, as the time of installation is a point that can act as an

orientation mark to describe the process. All stages before can and do influence each other; they can occur alternately or even in parallel. Additionally, we added external interrupts that reflect triggers from outside of the process itself. This could, for example, be a newly released update for the software during an update process. This might require a new deciding stage, potentially delaying the whole process. The area of external interrupts might be worth looking at in more detail as identifying those triggers, and their frequency could further help understand or even improve the process.

We do not imply that information, such as lessons learned during the installation, does not influence the post-deployment stage or the other way around for future updates. Li et al. [84] assigned all tasks that are deployment-related to the deployment stage. In our model, these fall into the *preparation* stage in pre-deployment. We argue that the preparation stage of our initial model, that includes the non-testing related preparation of the deployment process, is a more natural fit to the observed workflow.

The deployment itself is defined as the actual step of deploying the update on the live system, possible failures included. Since this task can fail due to various reasons (e.g., a different live- than staging-system that causes the patch to behave differently), there is a way back to the pre-deployment stage. Also, this step can differ vastly based on the scenario one observes.

As we observed tickets that ended in no deployment, we added an exit path that resembles the option of terminating the update process without deploying the update.

Once the update is successfully deployed, the *post-deployment* stage begins and includes every step after the installation. While the actions taken there could be modeled with more granular, we argue that for describing the update process itself, the pre-deployment stage is more important.

Ambiguous Actions

In the coding process, it was sometimes hard to decide between small nuances in the coding: a similar action can be coded as part of different stages depending on the context. For example, we had to decide how the search for failures during the installation has to be coded: should the code depend on the place where the search is done (e.g., on the testing system versus on the live system)? In the first case, this would fall into the testing stage, whereas in the latter, it would be coded as deployment. We learned that the best way to apply codes is based on the greater goal the action is aimed at. In the decision stage, this is ending up with a decision; in the preparation stage, it is being prepared to test and deploy; at the end of the testing stage, the goal is to know whether it worked and so on. Each action that mainly is focusing on reaching the goal was - in doubt - coded as part of the corresponding stage.

A word on self-reported vs. measured data

When conducting the interviews with all employees in 2017, only two project managers indicated to be involved in updates. However, when we analyzed the tickets from that year, three additional PMs took active roles in certain steps along the identified update process[3], e.g., opening a ticket to hint a developer to a new update, asking them to prepare the deployment. This can also be seen in Figure 4.1. While it is not surprising that people might not identify those small steps as involvement in updates, it again shows the necessity to be careful with self-reported data. Similar findings were already shown for various topics and user groups [135, 33, 108, 137].

4.4.1 Limitations

In the following section, we name and discuss the limitations of the study:

- **Case Study**: We studied one company in detail, and while we can be sure the aspects we found to be missing in the previous models were actually missing, we can't know if there other changes to the models would be needed to cover further aspects. More in-depth studies in other organizations are needed.
- **Complex data set:** The update process involves many stakeholders, different software types, and situations. Many tickets are similar on the high level, but most differ in some aspects. We pre-selected tickets of the data set to analyze the process: For example, we excluded the tickets that covered installations of the same update version for multiple projects. While these tickets give interesting insights into the processing of the deployment on multiple machines, this was not an area we focused on. We tried to analyze these tickets based on the process itself for each project, but in these cases, little information per update was given, and we sometimes could not distinguish the stage each project was in the specific notes.
- **False negatives:** We extracted update related tickets by looking for the appearance of the words "update" or "upgrade" within the tickets. This way, we might have missed issues that were update related but did not contain the two words. However, we got enough tickets to contribute to the model.
- **Missing stages:** In most of the tickets we analyzed, learning about an update and the decision to update was already made. Also, who would have to install the update was already decided most of the time. So in the analysis, these stages do not appear in the total number of transitions between the stages. So the distribution that is seen in Figure 4.3 has to be interpreted with this in mind.

[3]We double-checked that they worked at the company during the interview phase.

- **Omitted details:** The proposed model does not include every single possible action one can think of in the context of updating but is an abstraction of the process. The level of detail needed to further talk about the process may change over time.

4.5 Ethical Considerations

The interviews, the survey, and the export of the tickets were conducted by an employee of DevComp with their agreement. Since this study was conducted by the employee of the company, the University IRB was not responsible. Nonetheless, both the employee and we followed ethical best practices. We replaced all employee names, email addresses, company names of customers, and speaking names of projects and servers from the data. We did this in an automated fashion before the analysis, and during the analysis, we manually pseudonymized passages still containing sensitive data.

4.6 Summary

This chapter showed that the update process proposed in chapter 3 and by Li et al. [84] are not as flexible as needed. In the end, a new model emerged that hold this feature. It also takes external factors into account and enables future work to classify steps in the process better. The next chapter looks at another aspect that came up in the first study: update information. In a workshop paper, I once more observed administrators using interviews and a survey about their information sources and the information they need to make decisions.

Chapter 5

Update Release Notes

Disclaimer

This chapter's contents were previously published as part of the paper "What does this Update do to my Systems? - An Analysis of the Importance of Update-Related Information to System Administrators" presented at the 6th Workshop on Security Information Workers in 2020 [89] together with my co-author Florin Martius. As this work was conducted with Florin as a team, this chapter will use the academic "we" to mirror this fact. The idea and initial concept for this work came from me. Together, we designed the user-study. Florin Martius conducted the study, analyzed, and processed the results. Before compiling the paper for publication, we jointly discussed the study's implications.

5.1 Motivation

As already mentioned in the first study of this work in chapter 3, administrators rely on precise information about the update, for example, about dependencies, that help to decide whether and when to update. A lack of information hinders this learning phase and is a barrier to the update process [84]. Thus, a further investigation into the aspect of the provided and considered information is of interest.

Moreno et al. analyzed 1,000 release notes by hand. They stated that fixed bugs are the most frequent item included in release notes. Other standard information includes new code components, new features, and modified code components [95]. Abebe et al. observed three different styles in writing release notes: New features, bug fixes, and improvements [1]. By now, there are no standards [1] or guidelines on writing release notes.

In this chapter, we analyze which information administrators consider being necessary as part of their assessment. Therefore, we wanted to answer the following questions:

- Where do administrators obtain information related to updates?
- What information is relevant for the decision whether or not to update?

- How do administrators compensate for lack of information?
- What are the differences in handling security and feature updates?

The study revealed that release notes are the main source for learning about an update. When they are considered to be insufficient, the participants also referred to online forums and blogs. We identified the purpose, dependencies, and known issues as the most important information of release notes to system administrators. The study results also show that administrators reportedly install security updates in a far more timely manner than feature updates.

5.2 Qualitative Interviews

We wanted to understand how the information in release notes is processed by administrators. Therefore, we conducted five semi-structured interviews with system administrators from German companies. All of the participants were full-time administrators with more than 20 years of experience. We asked the participants (1) where they get informed about updates, (2) what information is relevant for the decision whether or not to update, (3) how they deal with a lack of information, and (4) about differences in handling security and feature updates. The interviews were conducted either over the phone or in-person and lasted between ten to 55 minutes. All interviews were recorded and transcribed by one researcher. The same researcher extracted key messages to virtual sticky notes, arranged similar statements together, and sorted them into groups. This resulted in the creation of affinity diagrams that can be seen in section C.2.

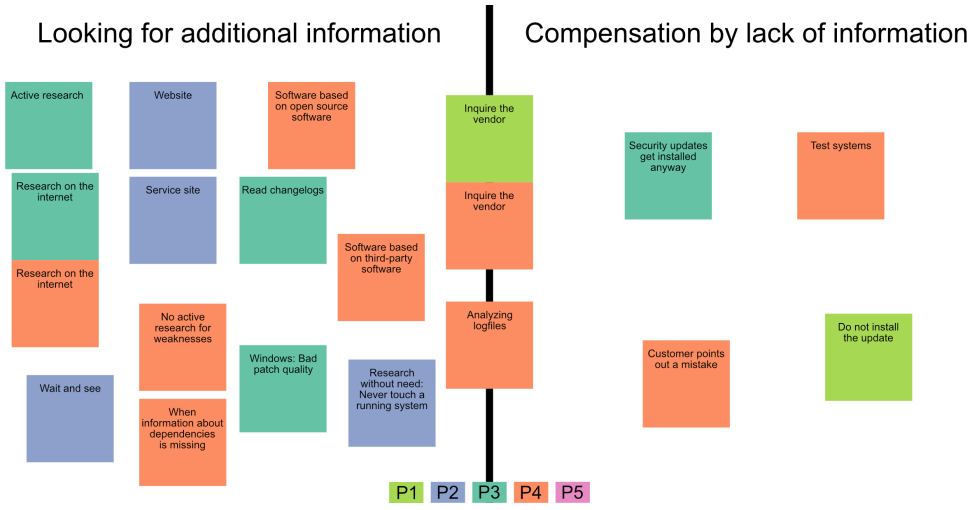


FIGURE 5.1: Affinity diagram of the answers about additional information sources and coping mechanisms in the case of missing information.

Figure 5.1 presents the results we gathered when asking the participants for the sources of their information, when they search for additional information (1), and how they cope with missing information (3). As a source of information, the internet was mentioned, alongside with the software itself (e.g. notifications) and reading of the change-logs. One participant mentioned that they wait some time before installing an update to see if other administrators faced any problems with the update. In case of missing information, two mentioned inquiring the vendor and one participant even refrains from deploying the update in some cases.

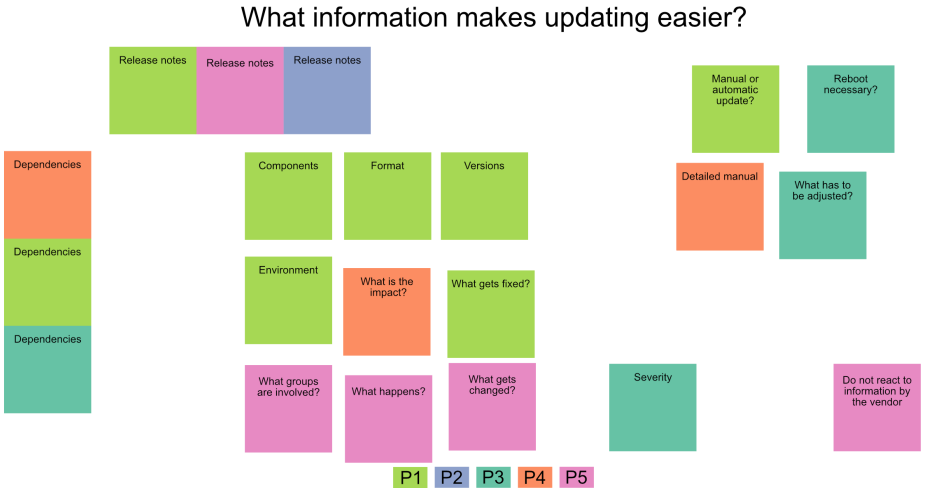


FIGURE 5.2: Affinity diagram of the answers about good and bad examples of information.

The diagram in Figure 5.2 shows the factors that help administrators in the decision process (2). Three of the five mentioned reading the release notes. Also, three participants take a look at the dependencies of the software that might be influenced. Besides this, other factors like the estimation of the impact and the changes or the information about a necessary reboot also came up which supports the findings of the work in the previous chapters.

In addition to this information, the answers to examples of good and bad information are presented in Figure 5.3. Things like a change-log, corresponding bug tickets (like in GitLab) or the information about the actual changes in the system (e.g., replaced files) are helpful for our participants. On the other hand, we gathered several examples that are considered as suboptimal, like missing, incomplete or incorrect information which can hinder the update process.

In alignment with related work, we found that there are obstacles for administrators to learn about updates. Four administrators reported a bad experience with past updates due to incomplete or incorrect release notes. All participants agreed that they install feature updates only when necessary. Before the installation, they want to know the purpose and the main changes to infer why this update is essential. Besides this fact, dependencies and requirements are key

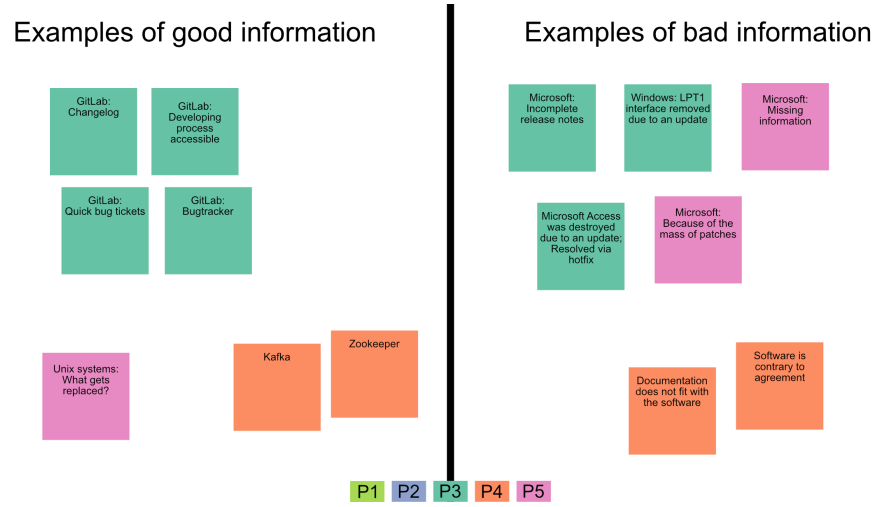


FIGURE 5.3: Affinity diagram of the answers about what information supports the admin in the decision to update.

information. In particular, P2 stated: *"I would include how the update should be installed, [...] the improvements [...], what it does and what was fixed. These three details are mandatory for an update. Unfortunately, they are not always included."*

All of the respondents mentioned that security updates get installed as soon as possible, contrary to feature updates that will only be applied if necessary. When the information provided within the release notes appears insufficient to the interviewees, they primarily search for information on the internet or contact the vendor.

5.3 Analysis of Update Release Notes

To determine what kind of information matters to system administrators, we wanted to understand which components can exist in update release notes. We therefore analyzed release notes of five broadly used software types that administrators have to deal with. Therefore, we picked that software the interview participants told us they are using. These were the Apache2 (web-server), Microsoft Windows, Red Hat Enterprise Linux, Debian (operating systems), and GitLab (version control software). We derived information from 15 release notes of those software and generated a classification based on the codes of Moreno et al. [95]. Table 5.1 presents the grouped types of information. A check indicates whether or not a release note of this vendor provides the associated information. As already obtained by Abebe et al. [1], no standards exist for writing release notes. In line with this, the analysis showed different approaches in providing update-related information: While some vendors like GitLab distinguish between security updates, bug fixes, and feature updates, others like Apache or Microsoft release unspecified updates containing security updates or bug fixes

as well as new implemented features. We observed that every release note contained a release number, and most of them contained the date the update was released and the purpose of the update. Changes in the environment were never stated, dependencies only once.

	Type	Apache (Unspecific)	Debian Feature/Security	GitLab Security	GitLab Feature	GitLab Patch	Microsoft Security	Microsoft Unspecific	Red Hat Security	Red Hat Feature	Red Hat Patch
General	Release Date		✓	✓	✓	✓	✓	✓	✓		✓
	Release Number	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Note Number										
	Note Date		✓	✓		✓	✓	✓	✓		✓
	Purpose of the Update		✓	✓	✓	✓	✓		✓	✓	✓
Summary	Fixed Bugs	✓		✓		✓		✓		✓	✓
	Still Existing Bugs										
	Steps to Reproduce Bug	✓				✓					
	Involved Components	✓	✓		✓	✓	✓	✓		✓	✓
	Changed Environment										
	Known Issues						✓	✓		✓	
	Closed Vulnerabilities	✓	✓	✓			✓		✓		
	Risk Qualification			✓			✓		✓		
	Added Feature	✓	✓		✓					✓	
Impact	Removed Feature	✓	✓								
	Modified Handling of a Feature	✓	✓		✓					✓	
	Advertising Information							✓			
Changes	Added Files	✓	✓			✓		✓		✓	
	Removed Files	✓	✓			✓				✓	
	Changed Files	✓	✓			✓	✓	✓	✓	✓	
Manual	Prerequisites		✓			✓	✓	✓	✓	✓	
	Dependencies		✓								
	Update Delivery						✓	✓		✓	
	Installation Manual itself	✓				✓	✓	✓	✓	✓	✓
	Third party										
Other	Documentation of Features	✓			✓					✓	
	CVE	✓	✓			✓	✓				
	Software Testing	✓		✓		✓					
	Disclaimers									✓	
	Support Contact Information		✓						✓		✓

TABLE 5.1: Classification of information and approaches of vendors.

5.4 Quantitative Survey

To quantify the importance of several information types, as seen in Table 5.1, we created an online survey based on our previous findings. As the results of the interviews suggest that well-written release notes can help system administrators

understand the impact of the update, we wanted to know what specific kind of information is relevant to system administrators. Therefore, we asked the participants to rate the importance of the different information types in the survey. After conducting the first survey in February 2020 with 41 participants, we improved the questionnaire and conducted a second survey with 16 participants in May 2020.

5.4.1 Structure

Both surveys consisted of four topics, of which the first three ones were based on the surveys of Li et al. and the one in the first study of this thesis. First, we asked about the participants' demographics, followed by a section about job-related information such as the company size or how long they worked as an administrator. Third, we asked general questions about update-related information that should answer which sources administrators use to collect information and how a lack of those pieces of information influences the update process. The last part of the survey aimed at obtaining how useful specific parts of update-related information are to the administrators. This part contained the types of information presented in Table 5.1 and was grouped by this classification.

We conducted a second survey because the first one revealed two areas of improvements that we wanted to investigate further: (1) First, to understand the differences in reading release notes between automatic and manual updates, we asked the participants to state how often they read release notes depending on the update type. Also, we added a slide bar where participants could state the percentage of automatic updates. (2) Second, we rephrased some questions and displayed the values of the answer options[1] of the Likert scales, to help the administrators rate the given statements. Additionally, we offered the respondents the option not to answer these questions. The final questionnaire can be seen in the section C.1.

5.4.2 Participants

We recruited the participants by personal contacts and link distribution on Reddit[2], Twitter[3] and Computerbase[4]. Before the survey was started, we presented information about the study's purpose to the participants and explained that their participation was voluntary and not compensated. The first survey was started 84 times, which resulted in 43 (51.2%) complete responses. We removed

[1] "Not useful at all", "Slightly useful","Moderately useful","Very useful","Extremely useful" instead of "1 - not useful at all","2","3","4","5 - highly useful"

[2] https://www.reddit.com/r/sysadmin/comments/gvw22r/study_survey_relevance_of_updatedrelated/, accessed: 06/19/20

[3] https://twitter.com/chrizzlz/status/1222463199833919488, accessed: 06/19/20

[4] https://www.computerbase.de/forum/threads/professionelle-systemadministratoren-fuer-studie-gesucht.1903976/, accessed: 06/19/20

incomplete responses. Two survey responses were excluded due to inadequate and false responses: One participant filled out the open-ended questions with nonsense answers; another stated having experience of 99 years by the age of 33. This left us with 41 valid entries.

Thirty-nine participants started the second survey, which led to 17 (44%) valid entries. After the sanitation of the data, we were left with a total of 58 completed questionnaires.

Survey #		1	2
n		41	17
Age in Years		20-60	18-58
	mn	34.75	30.41
	sd	8.95	11.04
Gender	Female	1	0
	Male	40	17
Location	USA	23	4
	Germany	9	10
	Other	9	2
Experience		0.5-25	1-25
in Years	*mn*	10.46	6.06
	sd	7.25	5.96
Company	IT-related	13	6
	Non IT-related	24	9
	Other	4	2
Company	$x \leq 10$	2	1
Size	$11 \leq x \leq 50$	5	2
	$51 \leq x \leq 100$	9	1
	$101 \leq x \leq 500$	16	7
	$501 \leq x \leq 2000$	0	2
	$x > 2000$	9	3
Administered	Clients	32 (78%)	13 (72%)
Systems	Servers	40 (98%)	15 (88%)
	Mobile	21 (51%)	6 (33%)
	IoT	7 (17%)	5 (28%)
	Other	6 (15%)	7 (39%)

TABLE 5.2: Demographic data of our participants.

Table 5.2 shows the demographics of the participants in both surveys. The age ranged from 18 to 60 years, with a mean of 33.5 years (sd=9.73). The population was mostly male-dominated (98%). All participants were located in Western countries: The majority lived in the US (27) or Germany (19). The remaining were spread over the UK (3), Canada (2), Argentina, Australia, Finland, the

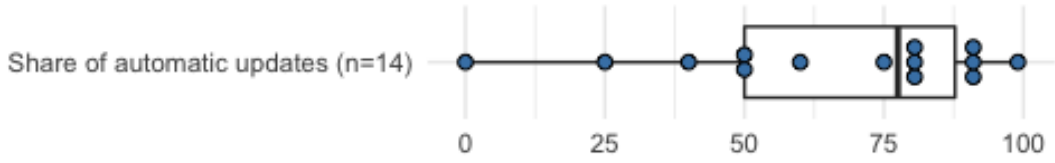


FIGURE 5.4: Relative share of automatic updates as stated by the participants.

Netherlands, New Zealand, and Switzerland (1 each). As stated before, we included a question in the second survey concerning the share of automatic updates, which the administrators face. This share ranged from 0% to 99% with a mean of 65.1% and a standard deviation of 29% as depicted in Figure 5.4.

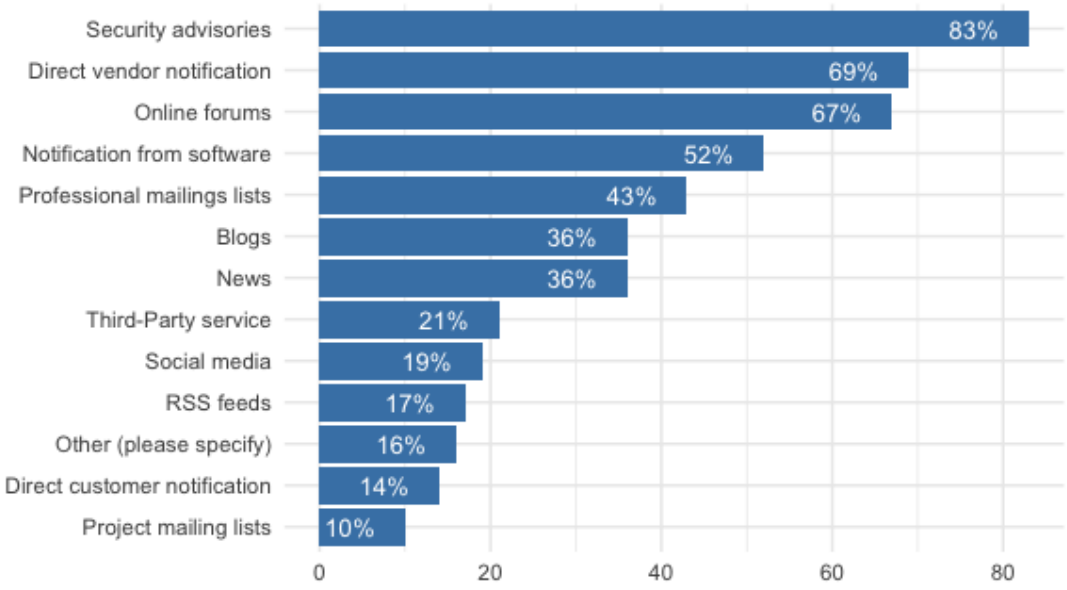


FIGURE 5.5: Sources of information reported by the participants ordered by the number of occurrences.

5.4.3 Results

We asked how much time the respondents can spend on learning about an update. The answers were divided into two groups of nearly the same size: While 47% of both surveys accumulated stated having no or too little time, 53% mentioned having sufficient time or more time than needed. Figure 5.5 shows that the participants reported that they mainly discover an available update by security advisories, direct vendor notifications, and online forums, which is in line

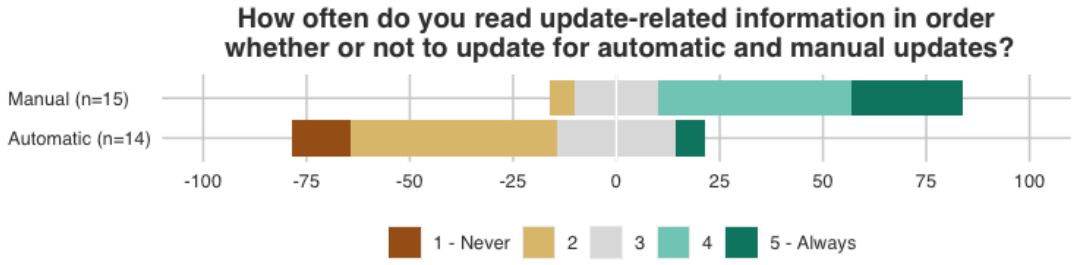


FIGURE 5.6: Overview of the responses to the frequency of how often participants read release notes on a 5-point scale from "1 - Never" to "5 - Always" based on the update type.

with the findings of Li et al. [84]. As seen in Figure 5.6, we observed that the respondents are not likely to read update-related information of automatic updates: 65% stated they never or rarely read them. In contrast, update-related information of manual updates is read frequently by the respondents: 72% stated they always or very often read them, 21% mentioned to do so sometimes.

Sixty-one percent of the participants stated that there is sometimes or more often a lack of information. Sixty-eight percent mentioned that a lack of information increases the effort to update. To compensate for missing information, 46% stated they always or very often look for additional information not given by the vendor. In this case, almost every participant (98%) uses online forums. Blogs (74%) and Security-advises (65%) were frequently marked answers, too.

The most useful information stated by the respondents were: The purpose of the update (95% in the first survey / 82% in the second), prerequisites (95% / 77%) and known issues (95% / 88%), followed by fixed bugs (91% / 70%), closed vulnerabilities and dependencies (85% each / 71% and 85%). In contrast, information that fewer than 20% specified as very or extremely useful are as follows: Disclaimers are identified as the least useful information, with only 11%/12% of respondents highlighting them as useful. Advertising information for the support level and the release note's date is mentioned second, with only 12%/20% of respondents marking them as a decision-making tool. Although, many respondents found the number of the release note to be less useful than the note date. Here, we also had participants who reported that the date is very beneficial (15%/36). Results of the entire types of information are listed in Table C.1.

5.5 Discussion

The study with system administrators identified that some types of information are more relevant than others. In this section, we will discuss and evaluate the results.

5.5.1 Implications

The results show that update-related information support administrators in the updating process. The survey indicates that the purpose and major changes, such as fixed bugs, are key information that coincides with the interview results. We infer that the administrators use these kinds of information to rate the urgency and update necessity. The following types of information useful for the respondents are dependencies and prerequisites to install the update. This suggests that administrators need to be aware of the requirements, like a mandatory restart, in advance to be able to schedule the deployment of the update. Similarly, missing necessary dependencies delay or even hinder the update process since the administrator must execute further steps like updating third-party software. This may explain why the study revealed that release notes of automatic updates are read rarely, contrary to manual updates: Automatic updates check dependencies and prerequisites automatically, so the administrator does not have to ensure to fulfill all requirements to install the update.

Known issues provide information about possible bugs that may occur after installing the update. The participants stated known issues as similarly helpful as the purpose or prerequisites of the update. However, they are a different kind of information than the update-related information stated before. They do not communicate intentional changes the update entails, and assessing these issues beforehand is hard. By knowing about bugs before they occur, the administrator can evaluate whether the bug might impinge the system and decide to update or refrain from the update until this issue gets fixed.

An update may impact the *support-level*, which means the administrator's handling with the software, or the *end-user-level*, which describes the end user's handling with the software. We observed differences in the usefulness of information related to those two handling levels. The respondents stated that changes on the end user level are more critical than on the support level. These results indicate that administrators are aware that end users do not like UI changes and want to prevent users from those.

As the study obtained that administrators install feature updates in a less timely manner than security updates, we follow the recommendations of [52, 84] in decoupling security patches from bug fixes or feature updates. This procedure has the advantage of allowing the administrators to close vulnerabilities without having to deal with undesired changes.

5.5.2 Comparison to End User Behavior

We identified several similarities and differences between administrators and end users in processing updates. As a similarity, P3 reported a bad experience with past updates, stating that a key feature was removed due to an applied update. The same frustration was found for end users who stated similar bad experiences with past updates [71, 132]. Also, the fact that some end users expect bugs in recently released updates [71] or wait a certain period before deploying the update due to expected bug fixes [131] could be observed in the interviews: P2 explained precisely the same method in dealing with feature updates. Similar to how all of the interviewees mentioned different handling between feature and security updates, Mathur et al. [91] observed that end users are more likely to install a security update than a feature update. Another similarity can be found in the way of gathering update-related information: Like almost half of the survey participants who stated that they look for additional information not given by the vendor, Vaniea et al. [131] found that some end users also searched for additional information, for example by consulting family and friends.

A noticeable difference is the general handling of updates. While several user studies observed that many end users did not understand the benefit of updates [52, 71, 90, 131, 132], all of the interviewees agreed that updating is important. This finding coincides with a comparison study between experts and non-experts, which has been conducted by Ion et al. [71], stating experts do know that updating is one of the best measures to maintain security. Mathur et al. [91] found that knowing the purpose benefits the update decision of end users. The results suggest that this is also the case for system administrators.

5.5.3 Limitations

The results rely on the self-reported data of the study participants. An administrator's update behavior depends on many factors, like, e.g., education, company size, or experience. As the surveys had only a small number of participants with non-representative demographics, the results are not generalizable to all system administrators. All interviewees were employed in German companies with more than 250 employees. The respondents of the survey are mainly located in the US or Europe. Also, we stress that a limited number of interviews cannot cover the whole spectrum of opinions. Besides, the recruitment strategy might enhance bias. For example, it should not surprise that participants recruited in online forums tend to use online forums as a source to gather update-related information. Due to the small sample of analyzed release notes, the analysis of update-related information is not complete.

5.6 Summary

In this chapter, I presented a study about the information sources and types that administrators take into account when deciding whether to update or not. This study showed that it could help setting up a well-defined frame when observing specific administrator-related tasks, out of which more graspable recommendations can be made. In the next chapter, I present another study focusing on a single task that is common for administrators: The TLS configuration of a web server.

Chapter 6

Related Work on TLS

For the next study in this work, I present related work about the Transport Layer Security (TLS) ecosystem, such as measurement studies and user studies related to its deployment or the effect of warning dialogues.

When correctly deployed, Transport Layer Security (TLS) [31] protects the integrity and privacy of digital communication. However, different TLS features and protocol versions have been shown to have vulnerabilities, thus making several configurations (i.e., combinations of such features) insecure [26]. BEAST and DROWN are examples of effective and practicable attacks against TLS [15, 66]. To understand the real-world vulnerabilities of the TLS ecosystem and the diversity of TLS (mis-)configurations, researchers examined TLS deployments in measurement studies and user studies.

6.1 Measurement Studies

Internet-wide scanning tools, such as ZMap [35] and Censys [36], are used to measure TLS in the wild. They were used in studies that identified frequent configuration problems that potentially lead to browser warnings and create attack surfaces [8, 24, 37].

Ouvrier et al. [102] passively monitored 232 million HTTPS sessions and reported that more than 25% of the sessions had weak security properties. Gustafsson et al. [60] analyzed differences in public Certificate Transparency (CT) logs, while Holz et al. [67] evaluated the security of email and chat infrastructures, and reported "a worryingly high number of poorly secured servers". With the recent evolution of smart environments, new TLS-secured device classes have popped up. Samarasinghe and Mannan [113] measured the TLS parameters of 299,858 devices (e.g., cameras), and the authors found that such devices are usually more vulnerable than the Alexa Top Million sites. Common security problems included the use of RSA 512-bit keys, the RC4 stream cipher, or SSLv2 and SSLv3. Finally, Van der Sloot et al. [130] compared different measurement approaches and found that comparative analyses using aggregated CT logs, Censys snapshots, and Alexa 1M scans provide accurate snapshots of the TLS ecosystem.

Durumeric et al. [38] tracked the vulnerable population after the disclosure of Heartbleed, and found that, even after two days, 11% of the Alexa 1M sites remained vulnerable. Popular sites responded more quickly, while 3% of the analyzed population remained vulnerable as long as two months after being notified.

Kranch and Bonneau [77] investigated the use of novel security features such as HTTP Strict Transport Security (HSTS) and public-key pinning, and identified usability problems as the main reasons for reluctant upgrade behavior. The authors reported that "even conceptually simple security upgrades [are] challenging to deploy in practice." Amann et al. [13] claimed that only the Signaling Cipher Suite Value (SCSV) and Certificate Transparency "have gained enough momentum to improve the overall security of HTTPS.".

6.2 User Studies on TLS

Most TLS-related user studies focus on end-users, and their reactions to warnings. Sunshine et al. [119] conducted the first lab study examining the efficacy of current browsers' TLS warnings and evaluating two custom warning designs. Harbach et al. [63] studied how aspects of a warning message influence user reactions and found that linguistic properties have a strong impact. Several other studies were performed in the lab [117], online [49], and in the field [48, 49] to analyze the impact of the warning design and contextual factors [111] on users' click-through rates, and found that better warning designs can increase adherence rates [49].

Compared to the wealth of research focusing on end-users, there is far less focused on administrators. Fahl et al. [45] surveyed 755 web developers and investigated the reasons for deploying non-validating X.509 certificates on publicly available websites. Although one third of the participants admitted to misconfiguring the web servers accidentally, the majority stated that they knew about the problem, and gave reasons for their configuration choices. For example, some system administrators mentioned the high prices of CAs as a reason for intentionally deploying non-validating certificates; others stated that they did not trust CAs or had trouble configuring virtual hosts. Based on a mental model study by Krombholz et al. [80], administrators lack of conceptual mental models of HTTPS.

Schechter et al. [118] conducted user studies where the authors compared the effect of role-playing in studies on the outcome. They showed in a phishing study with end-users that participants in the role-playing scenario behaved significantly less secure than those who faced a more realistic one. Komanduri et al. [76] also compared a survey to a scenario-based task description and found that users tended to choose better passwords in the latter scenario.

Chapter 7

A Usability Evaluation of Let's Encrypt and Certbot

Disclaimer

The contents of this chapter were previously published as part of the paper "A Usability Evaluation of Let's Encrypt and Certbot: Usable Security Done Right" presented at the 26th ACM Conference on Computer and Communications Security (CCS) in 2019 [124] together with my co-authors Emanuel von Zezschwitz, Maximilian Häring, Katharina Krombholz, and Matthew Smith. As this work was conducted with my co-authors as a team, this chapter will use the academic "we" to mirror this fact. The idea and initial concept for this work came from me. The user-study was designed by Matthew Smith and me and conducted by Maximilian Häring and me. Katharina Krombholz provided helpful information about their study on which we built our work. Analyzing the study results was joint work with Emanuel von Zezschwitz and Matthew Smith. While I analyzed the quantitative part, we coded the support channel messages together. Before compiling the paper for publication, Emanuel von Zezschwitz, Maximilian Häring, Matthew Smith, and I jointly discussed the study's implications.

7.1 Motivation

Transport Layer Security (TLS) is among the most important protocols to secure data in transit, and has been an active research topic in the usable security domain, especially regarding the end-user's perspective, e.g., [119, 49, 111]. For a decade, substantial effort has been invested in improving the efficacy of TLS warnings. From one of the earliest works by Sunshine et al. [119] to today, usable security researchers have attempted to find ways to help end-users make good decisions when faced with such warnings.

However, end users are only one part of the picture. Akhawe et al. conducted a large-scale measurement study [11] and estimated that end-users would

see 15,400 false positive warnings per true positive warning due to server misconfigurations.

In 2015, Let's Encrypt (LE) began operating, to increase TLS adoption by offering free certificates. Let's Encrypt is a non-profit certificate authority (CA) that was founded "to reduce financial, technological, and education barriers to secure communication over the Internet" [9]. In conjunction with LE, the Electronic Frontier Foundation (EFF) offers Certbot, a tool that automates the acquisition and configuration of LE certificates for web servers [41]. The hope of this initiative is to reduce the barriers and improve the usability of the TLS setup. The data published by LE suggests that adoption rates are rising [83], and that it is mainly impacting the lower-cost end of the web, as 98% of the LE certificates are issued for domains outside the Alexa 1M [10].

Manousis et al. [87] found that only 50% of the domains that obtained an LE certificate actually responded with a valid LE certificate on the standard HTTPS port. The authors concluded that despite the many positive effects of LE, "there are serious misconfigurations among many website owners who use Let's Encrypt".

To shed light on where the adoption problems above stem from, and to examine the advantages of LE, we conducted a **randomized control trial to compare the usability of the EFF's Certbot with the traditional certificate configuration approach**. The contributions of this paper are as follows:

1. We present a quantitative study with 31 computer science students that compares the usability of two different methods for interacting with a certificate authority (CA) and configuring TLS on a web server.

2. We show that Certbot's usability improvements are particularly important for lower-skilled participants.

3. We analyze in which areas the automation of Certbot is particularly important.

4. We discuss what lessons can be learned from Certbot and identify areas where these do not apply easily.

5. We provide a methodological discussion of conducting lengthy laboratory user studies with expert users, such as administrators, and share the lessons learned.

The two relevant works to our research is 1) Krombholz et al.'s [79] and 2) Bernhard et al.'s [19] user-study.

The present study is an extension of the study protocol used in Krombholz et al.'s user-study on the deployment process of HTTPS. They conducted an observational lab study with 28 knowledgeable users in which they simulated a simplified certificate acquisition and standard deployment process. The study used a minimal web-based CA where participants could acquire TLS certificates

to be manually installed on an Apache web server. The study revealed a host of usability issues that often resulted in vulnerable configurations. The study did not contain conditions in which participants used LE and Certbot. The study also did not inform participants about which security requirements they should meet. Contrary to this, the following study differs in several ways. First, we conducted a randomized control trial to compare a traditional CA approach to Let's Encrypt and Certbot. We also explicitly told participants which security goals should be reached, and how the security of the resulting configuration could be evaluated. We made this change because Naiakshina et al. found that computer science students did not add security unless explicitly asked to [98]. The final important difference in the study design is that we formalized the interaction between the experimenter and the participants. In the Krombholz et al. study, technical assistance was given; however, this was done in situ, and was not planned in advance. In addition, the help was not recorded, and it was not analyzed. We created a Mattermost support channel for in-study realism, as well as to deliver consistent and recorded interaction with the participants. Our records on when participants required which kind of help offer valuable insights into the usability challenges. A final important difference concerns the participant sample. Krombholz et al. invited the 30 best students of the pre-screening survey of whom 28 participated in the study. We did not filter out lower-skilled participants because we wanted to see the effects of Certbot on different skill levels.

The other relevant work is that of Bernhard et al. [19] which appeared shortly before this one. They analyzed the usability of Let's Encrypt in comparison to a traditional CA approach. They conducted two studies: one within subjects with nine participants and one between subjects with ten participants (five per condition). In the first study, none of the nine participants managed to complete the traditional CA task, and only four managed to complete it with Let's Encrypt. In the second study, the authors got conflicting information. In this study three of five participants managed to complete the configuration in each condition. The authors stated that this was likely due to a change in recruitment criteria which was introduced in the second study to raise the skill level of the participants. Due to this, and the small sample sizes, the authors stated that they had found no reliable effects, and even conflicting information on which system offers better usability. In conclusion, they wrote: *"However, we did not find conclusive evidence regarding which method [Let's Encrypt vs. Traditional CA] 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."* The present study has a larger sample size, so it does not suffer from these issues. The study also gathered additional details via logging and the Mattermost support channel, so the analysis can go into more detail about where participants faced challenges and how Certbot helped them.

The additional related work on this topic can be found in chapter 6.

7.2 Research Questions

The research questions are split into two groups. The first relates to the main subject matter, the usability of Certbot.

- **Does Certbot support its users in fulfilling the task of enabling TLS?** Related work has shown that users struggle with manually deploying SSL certificates. We want to measure Certbot's performance and capability to help administrators set up TLS correctly compared to the manual approach, to quantify the performance, as well as to draw lessons learned from the Certbot approach.
- **How do participants perceive Certbot's functionality and usability?** Although automated configuration has many usability benefits, it is an open question whether administrators feel comfortable with the decreased level of control they might perceive due to automation.
- **How can the Certbot process be improved?** Although Certbot has a reputation for good usability, we are interested in possible areas of improvement, to support even more users in deploying secure TLS correctly. Because the usability is likely to be good from the start, we do not expect major improvements, but are open to the possibility.

The second group of questions relates to study methodology for administrator studies. Usable security researchers have a decade of experience in end-user studies. Studies with developers and administrators do not have the same body of knowledge yet. Naiakshina et al. found that the way tasks are framed for computer science students and freelance developers has a significant effect on how participants deal with security [98, 96, 97]. To add to this body of knowledge, we introduce the following research question:

- **How does task framing affect how participants behave in the study?** A common method used to elicit realistic behavior in end-user studies is to use a role-playing scenario [118, 76]. We are interested in seeing whether this tool is also useful for studies with experts like administrators or developers who are represented in this study through student proxies [98].

7.3 Methodology

7.3.1 Study Design

Similar to Krombholz et al. [79], we opted for a lab study to monitor and control the participants' behavior. In contrast to Krombholz et al.'s study, we looked at two independent variables. We conducted an A/B test to compare the usability of Certbot with a traditional CA. Thus, we had two treatment conditions:

"CA-Certbot" (CA-Cbot) for the Certbot with Let's Encrypt condition and "CA-Traditional" (CA-Trad.) for the traditional manual CA approach. Although it would have been nice if we could have used the same web CA as used by Krombholz et al. to enable a more direct comparison with their work, we opted to use a more complex one that resembles the realistic workflow of acquiring a certificate from an existing CA. In particular, the method included ownership verification. There, a server owner has to prove that they are in possession of the server and domain by placing a specific file in the web folder or by responding with defined content to a request made by the CA. We opted for these improvements because they would give a fairer comparison for the CA-Certbot condition which has the full complexity of the real-world implementation. Because we assume that the configuration task is highly dependent on personal skills, we opted to study the two conditions within subjects, because the sample size which would have been needed to balance out personal skill in a between-subject design would have been unattainably huge. To counter learning and fatigue effects, we randomized the order of conditions: Half the participants were assigned to use CA-Traditional first, and the other half started with CA-Certbot first.

The second variable is a meta-variable concerning the task framing. In a developer study conducted with students, Naiakshina et al. reported that in post-task interviews, some participants excused poor or no security performance by stating that they would have tried harder if they had been working for a real company as opposed to participating in a study [98]. This is a general problem for security-focused user studies in which participants know they are taking part in a study. There is always the risk that participants behave less securely because they know they are safe in a study environment or that they behave more securely because they want to impress the experimenters. A possible approach to mitigate this problem in end-user studies is to construct a role-playing scenario, and make the task as realistic as possible, to get participants into the "right" frame of mind. However, because we do not have the body of experience with expert studies that we do with end-users, it is not clear whether this kind of role-playing is necessary or beneficial. Therefore, we opted to introduce a variable to study the effect due to framing as well. For half of the participants, the task was framed as a study task (*Framing Study*); i.e., study-related user names (e.g., HXR) and passwords (e.g., HXR12345) were used. For the other half of the participants, we created a role-playing scenario (*Framing Role-Play*) in which they were asked to imagine they were working for a company. Thus URLs, user names, and passwords were tailored to be realistic. Naturally, such a framing variable cannot be studied within subjects, but has to be studied between subjects.

The four conditions we used in the mixed within-/between-study design can be seen in Table 7.1. The effects of the different configuration conditions CA-Certbot and CA-Traditional were evaluated within subjects while "framing" effects were evaluated between subjects.

		Between: Framing	
		Role-Play	Study
Within:	CA-Cbot	1: CA-Cbot+RP	2: CA-Cbot+Study
CA	CA-Trad.	3: CA-Trad.+RP	4: CA-Trad.+Study

TABLE 7.1: The four conditions we used in the study.

After completing each configuration task, the participants filled out an online survey that asked them about several aspects of the tasks they had performed, e.g., their self-assessment of their performance and their perception of the difficulty. After completing both tasks and the questionnaires, a final questionnaire was presented which directly compared the CA-Certbot and CA-Traditional tasks. The questionnaires can be found in section D.1 and section D.2.

7.3.2 Task Design

To tie the findings to related work, and to allow for a better comparison, the task design was based on the study by Krombholz et al., with some modifications as described in this section. The main task of the lab study was to acquire a certificate for a remote Apache web server and configure HTTPS with clear security expectations. Figure 7.1 shows the workflow scheme of the TLS configuration process from Krombholz et al.'s study that includes nearly all steps that are technically necessary in the manual approach, and that is similar to the CA-Traditional condition. To illustrate the Certbot automation approach, we enclosed the steps that Certbot automates with a grey box in Figure 7.1.

Sub-task 1: Baseline (*SSH* and *Apache admin*).

Sub-task 1 consisted of logging on to the study server using SSH and executing some basic copy commands to place some web pages in the www directory of Apache. Sub-task 1 was used as a non-security baseline to see if participants had basic Linux skills. If participants failed in this task, their performance on the other tasks had to be taken in the context of their low Linux skill level. These two steps will be referred to as *SSH* and *Apache*.

Sub-task 2: Certificate Acquisition (*CA*)

This sub-task included the steps "Create keypair & CSR[1]" and "Interact with CA" of Figure 7.1. We had the A/B test between CA-Certbot and CA-Traditional. In the CA-Certbot condition, participants were told to use Let's Encrypt to acquire and install a certificate. In the CA-Traditional condition, participants used

[1]Certificate signing request

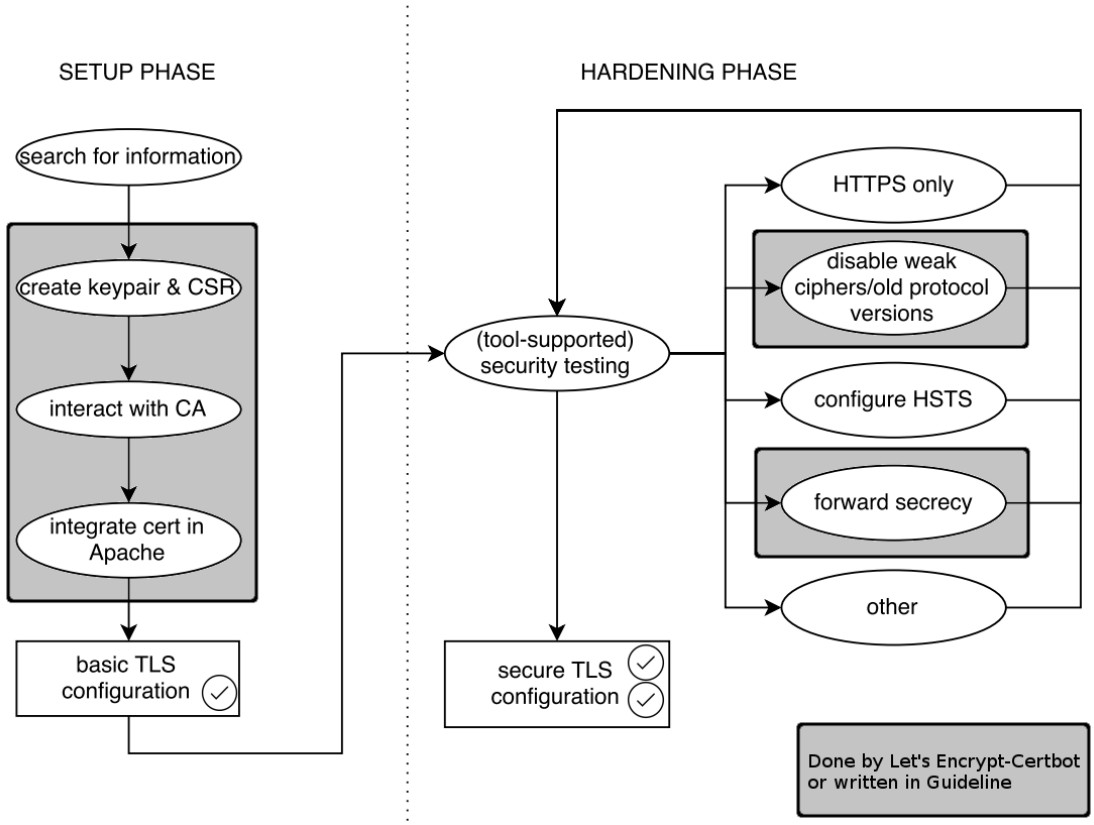


FIGURE 7.1: The workflow scheme of Let's Encrypt based on Krombholz et al. [79]

a traditional CA to acquire a certificate. Krombholz et al. used a custom minimalistic CA which did not resemble the user experience of a real CA. To make the traditional CA condition (CA-T condition) more realistic, we provided a forked version of *gethttpsforfree*[2]. This website resembles the steps a website administrator has to take for several official CAs, such as Comodo[3] and provides a guideline.

Sub-task 3: Configuration (*Conf*)

In this sub-task, we had the A/B test between CA-Certbot and CA-Traditional, insofar as in the CA-Certbot condition acquisition and installation could be combined, and in the CA-Traditional condition, the participant had to manually install the certificate acquired in sub-task 2. This task resembled the "Integrate cert in Apache" phase in Figure 7.1.

[2]https://gethttpsforfree.com, Accessed: 02/06/2019
[3]https://secure.instantssl.com/products/SSLIdASignup1a, Accessed: 02/06/2019

Sub-task 4: Configuration tests

The study by Krombholz et al. ended after sub-task 3 and evaluated what participants submitted, based on criteria not known to the participants in advance. As stated, Naiakshina et al. found that students did not implement any security in a study setup unless specified to do so. Therefore, we specified the security requirements in the task description and added an explicit sub-task in which participants were asked to check their configuration using the "Qualys SSL Server Test" tool[4] Krombholz et al. used to evaluate the results for those participants. The details are presented in section D.5.

Timeframe

Due to the within-subjects design, each participant completed the configuration task twice, once with each approach. To avoid the study seeming tedious and fatiguing, we wanted to keep it as short as possible, while at the same time allowing enough time that participants could realistically complete the tasks. To determine the time needed, we conducted several pre-studies, and settled on a maximum editing time of three hours for the CA-Traditional task and a maximum of two hours for the CA-Certbot task. After the time limit was exceeded, the participant was asked to continue with the next condition. The observations from the pre-study suggested that if participants had not solved the tasks within these time limits, they would not be able to complete the task within the study context. Thus, we counted the participants as failing that task without making excessive demands on their time.[5]

7.3.3 Participants

One particular challenge for conducting studies with experts is acquiring a satisfactory number of participants. Therefore, we conducted the study with computer science students, because recruiting enough professional administrators for a five-hour lab study was not feasible at this stage. There is also a growing body of evidence that computer science students can serve as proxies for administrators and developers in user studies [79, 143]. In particular, Naiakshina et al. found that students are viable proxies in a password storage study in which the authors compared students to freelance developers [97]. Thus, although computer science students are not exactly the same type of user as professional administrators, we believe that they are acceptable proxies for the A/B study we conducted.

[4] https://www.ssllabs.com/ssltest/ Accessed: 09/02/2019

[5] In retrospect, it would have been better to give CA-Certbot the time as CA-Traditional even though it was not necessary for CA-Certbot itself. We discuss this point in the limitations section 7.5.

7.3.4 Recruitment and Demographics

For the first pre-study, we recruited three participants known to our group who had experience in usability studies. These participants gave feedback on the early study design.

We then recruited participants using a survey distributed via the computer science mailing list of our university. The survey was based on Krombholz et al.'s work [79] (see section D.3). Sixty-eight participants filled out the questionnaire. Ten participants who did not fill out the questionnaire completely were removed from the selection process. We invited all 58 remaining students to participate in the lab study. Forty-five participants responded to the invitation, and 38 actually took part. Krombholz et al. found that previous experience in configuring web servers is a predictor of success. To avoid this becoming a confound, in particular because we did not exclude students with less experience, we ranked the participants based on two criteria: 1) whether they had previously configured a web server and 2) the number of correct answers in a pre-screening questionnaire. This ranking was used to build pairs of students with similar experience who were then randomly assigned to one of the two framing conditions, "Framing Study" and "Framing Role-Play". Assignment to the CA conditions was alternated.

We conducted a second pre-study with four participants (one in each condition) to further test and improve the experimental design. This left us with 34 participants who completed the main study.

Three participants were removed from the data set: One participant completed the first task (CA-Traditional) twice instead of each task once, and one participant successfully completed the first task (CA-Certbot) but left the study without attempting to complete the CA-Traditional task. Another participant encountered technical problems due to a temporary bug in the Certbot repository. Table 7.2 shows the demographics of the remaining 31 participants.

All participants were compensated with 80 Euros. We received IRB approval for the study. All participants consented to the study and signed a written consent form.

7.3.5 Support Channel

The main goal of the study was to compare the usability of the CA conditions (CA-Traditional and CA-Certbot), and identify common pitfalls and potential areas of improvement. Several issues complicated this goal. First, it was important to distinguish between usability problems of the CA system and the general technical difficulties that participants might encounter. Related user studies with complex tasks showed that there is the risk of a participant failing early on, and thus, never getting to the tasks of interest [140]. Second, in relatively long procedures, such as in this study, simply asking participants to report problems at the end of the experiment runs the risk of participants forgetting some of the

Demographic	Number	Percent
Gender		
Female	3	10%
Male	28	90%
Age		
Min.	18	
Max.	34	
Median	25	
Experience as sysadmin		
Yes	22	71%
No	7	22%
No answer	2	7%
Configured TLS before		
Yes	15	48%
No	16	52%
Currently employed as an administrator		
Company web server	3	
Private web server	1	
Non-profit organization web server	9	

TABLE 7.2: Participants' demographics ($N = 31$)

problems they had. It is especially likely that big problems mask smaller problems when participants recall the problems after the task.

To counter this issue, we introduced an in-scenario support channel, similar to the study pilot used by Garfinkel et al. to interact with participants [57]. We used the Mattermost chat client[6], an open source web chat platform, and a playbook (see section D.4) to implement the support channel. Mattermost was pre-installed on all machines, and participants were told that they could message two contacts listed under "direct messages" named *support* and *supervisor* if they encountered any problems that they could not solve on their own.

This support channel offered several benefits. First, if participants had non-CA-related difficulties, e.g., while using SSH to connect to the server, or setting permissions for copy operations, we were able to provide assistance, so that the participants were able to proceed with their main task. The fact that assistance was requested was noted, and was included in the evaluation. Second, we received feedback at the moment when problems occurred. Similar information could have been acquired using the think-aloud method, but we opted for the in-scenario channel to avoid the well-known awkwardness of the think-aloud protocol. In addition, there have been reports that think-aloud does not work well in long developer studies [98].

[6]https://about.mattermost.com/ Accessed: 02/06/2019

To ensure that the support channel would not be used inconsistently, the experimenter had to strictly adhere to the following procedure.

1. If the question could be answered by referring the participant to the task description, this was done.

2. If the question was a general technical question, and equally applicable to both CA conditions, help was given, and a note was made.

3. If the question was directly related to a CA aspect of the task, the experimenter remotely analyzed what participants had done up to that point and then made the following judgment call: If the experimenter had the impression that the participant had not tried hard enough or was close to finding a solution without further help, the experimenter would respond to the participant about 10 minutes after their message to help. In addition to the couple of minutes needed to check the participant's actions, this delay was designed to raise the threshold for participants to use the support channel.[7]
   If this kind of support was given, the following levels were used:

   (a) If possible, only a nudge was given. This nudge would not solve the problem but point the participant in the right direction to solve the problem without further help.

   (b) If that was unfeasible, a hint was given that would solve the specific problem; e.g., the experimenter pasted the required command in the chat, similar to how normal support staff operate.

   (c) And if that was unfeasible, the experimenter completed a sub-task for the participant, e.g., sending the CSR, sending the signed certificate, or installing the Certbot.

The last two options were last resorts. These sub-tasks were then marked as failures for the participants because they received CA-specific support. All other encounters fell into the category non-CA-specific support. Both categories are defined in more detail in subsection 7.4.4.

7.3.6 Technical Setup

The study was conducted in our usability lab which can hold up to eight participants at the same time. Each participant had a workspace with a computer running an installation of the study OS based on Ubuntu. Each participant had a set of over-ear noise canceling headphones. We also provided an overview sheet

[7] This option turned out to not be needed, and we never had to wait 10 minutes. There was one support request which the participant solved without help even before we could have answered. In all other cases, there was ample evidence that participants had tried to solve the problem on their own first.

with credentials for Mattermost and Ubuntu, and a text describing the structure of the study. An example can be seen in section D.5. The Ubuntu desktop was empty except for a link to the Mattermost chat client. The web server to be configured was running on an Amazon AWS server reachable via the domain given in the task description. Apache2 was already installed with the default configuration.

No special restrictions were introduced for the handling of the computer or the external server running the web server. The participants were equipped with root access on the server. After the task was completed, the image of the computer was automatically saved, along with the browser history, the bash history, and the Apache configuration files. Screen capture software recorded the entire procedure for the task.

7.4 Results

In this section, we present the results from the lab study. We conducted qualitative and quantitative analyses. Qualitative data was collected from analyzing the discussions of the communication channel, as well as free text answers in the survey data (answered after each condition). Quantitative data was gathered from the analysis of the screen recordings of each participant in combination with the collected bash log files and the Apache2 configuration files. Unless stated otherwise, analyses were performed on the 31 participants who were exposed to both CA conditions. We found no significant differences concerning the framing variable. Therefore, the following analysis focuses on the CA variable.

7.4.1 Task Completion

In the following, we present the study participants' success rates, as well as the reasons for failure, as can be seen in Table 7.3 and Table 7.4. Please note that it is possible for a participant to fail at a single task and still continue on, so each column represents the local view of that step. All 31 (100%) participants succeeded in the SSH task in both conditions. Twenty-eight (90%) successfully deployed the website documents in the CA-Certbot task and 29 (94%) in the CA-Traditional task. These were the two tasks we used to judge basic Linux/server configuration skills. Twenty-eight (90%) participants in CA-Certbot and 23 (74%) participants in CA-Traditional successfully interacted with the CA and acquired a valid certificate. Twenty-eight (90%) participants managed to correctly deploy the certificate with Let's Encrypt, 16 (52%) using the traditional approach.

All 28 participants in the CA-Certbot and 29 participants in the CA-Traditional condition who got to the CSR stage succeeded in creating a CSR. At that point different problems occured. To dive deeper into the results, Table 7.5 provides

	CA-Certbot				CA-Traditional			
	SSH	Apa	CA	Conf	SSH	Apa	CA	Conf
P1	✓	✗	-	-	✓	✗	-	-
P2	✓	✓	✓	✓	✓	✓	✓	✓
P3	✓	✓	✓	✓	✓	✓	✓	✓
P4	✓	✓	✓	✓	✓	✓	✓	✓
P5	✓	✗	✓	✓	✓	✓	✓	✗
P6	✓	✓	✓	✓	✓	✓	✓	✓
P7	✓	✓	✓	✓	✓	✓	help	✗
P8	✓	✓	✓	✓	✓	✓	✓	✓
P9	✓	✓	✗	-	✓	✗	-	-
P10	✓	✓	✓	✓	✓	✓	✓	✓
P11	✓	✓	✓	✓	✓	✓	✓	✗
P12	✓	✓	✓	✓	✓	✓	✓	✓
P13	✓	✓	✓	✓	✓	✓	✓	✓
P14	✓	✓	✓	✓	✓	✓	✓	✓
P15	✓	✗	-	-	✓	✓	✗	-
P16	✓	✓	✓	✓	✓	✓	✓	✗
Sum	16	13	13	13	16	14	12	9

TABLE 7.3: Overview of the participants who started with Let's Encrypt. In both conditions, the following sub-tasks had to be executed: *1) SSH-Connection to the web server, 2) Configuring Apache, 3) Acquiring a certificate from the CA, 4) Configuring the web server to serve the certificate*. "✓" symbolizes a success, "✗" a failure at this step and "-" means that the participants did not even start this sub-task.

an overview of the certificate-related steps and problems (occurrences denoted by numbers in braces).We divided the table into four sub-groups:

- *Certbot*: In this step, the user has to install Certbot using the operating system-dependent repository and start it.
- *CSR*: Then, the user has to create a key pair that is used to create a CSR. In this step, they have to choose the key size and the hash algorithm. They also have to decide for which domains the certificate should be valid and create the actual CSR.
- *Prove ownership*: In this step, the user must prove that they are in control of the domain for which the certificate will be issued. To do that, she must host a specific file on the server the domain is pointing to. After the successful ownership verification, the certificate is generated and provided.

	CA-Traditional				CA-Certbot			
	SSH	Apa	CA	Conf	SSH	Apa	CA	Conf
P17	✓	✓	✓	✓	✓	✓	✓	✓
P18	✓	✓	✓	✓	✓	✓	✓	✓
P19	✓	✓	✓	✓	✓	✓	✓	✓
P20	✓	✓	✓	✗	✓	✓	✓	✓
P21	✓	✓	✗	-	✓	✓	✓	✓
P22	✓	✓	✓	✓	✓	✓	✓	✓
P23	✓	✓	✓	✗	✓	✓	✓	✓
P24	✓	✓	✓	✓	✓	✓	✓	✓
P25	✓	✓	hint	-	✓	✓	✓	✓
P26	✓	✓	help	✓	✓	✓	✓	✓
P27	✓	✓	✓	✓	✓	✓	✓	✓
P28	✓	✓	✓	✗	✓	✓	✓	✓
P29	✓	✓	✓	✗	✓	✓	✓	✓
P30	✓	✓	✓	✓	✓	✓	✓	✓
P31	✓	✓	help	✗	✓	✓	✓	✓
Sum	15	15	11	8	15	15	15	15

TABLE 7.4: Similar to Table 7.3, the overview of the participants' sub-task success. This table displays the data of the participants who started the study with the traditional CA.

- *Certificate Installation*: Now, the user has to integrate the certificate in the Apache2 web server, enable SSL, create a config file, and enable the site. As an option, they can continue with a hardening phase.

For each of the steps, we highlighted in what areas knowledge or skill is useful for that step. We differentiate between three areas: 1) Apache. For these steps, skills in configuring Apache are needed. 2) Operational. Knowledge about the operating system and how the system is to be used in the end is needed. In this case specifically, it is knowing which domains are to be used. 3) Security. In these steps, users are exposed to security concepts and have to interact with security tools. Black circles indicate areas where a lack of knowledge or skill could lead to failing the step.

A surprising finding in our view is that the security or CA aspects did not seem to cause the participants trouble. Instead, the steps in which participants needed knowledge or skill to configure Apache were difficult.

Three participants struggled with the ownership verification, where they needed to configure the server to host a specific file at a defined URL. They could not manage to configure this so that the CA could verify ownership. Two participants had problems deploying the certificate on the web server due to the UNIX file and permission system that, e.g., prevented them from copying files. These problems seem to be problems with the handling of UNIX, Apache2, and

Step	Area	Info				
	Apache2 Operational Security		CA-Certbot	Failed	CA-Traditional	Failed
Certbot						
Install	○●○		M	-	not necessary	-
Run	○●○		M	1	not necessary	-
CSR						
Create key pair (public+private key)	○○●	key size & algorithm	A	-	M	-
Define domains	○●○		M	-	M	-
Create CSR with domains	○○●	key size & algorithm	A	-	M	-
Prove ownership						
Serve file at specific location on web server	●○○		A	-	M	3
Certificate Installation						
Deploy certificate	●○●	file permissions	A	-	M	2
Enable Apache2 SSL module	●○○		A	-	M	1
Create SSL configuration file	●○●	ciphers & protocols	A	-	M	4
Enable site	●○○		A	-	M	1

TABLE 7.5: A detailed view of the steps and challenges of the CA and Configuration task. Beneath each step, the corresponding type of knowledge is mentioned that is needed to execute it. An **"M"** in the right columns indicates that this step has to be performed manually; **"A"** means that this step is automated.

bash, and are not directly security tasks. However, they are necessary for the configuration.

Another participant did not know that the SSL module of Apache2 has to be enabled to serve websites over HTTPS. One problem occurred because the participant created a new configuration file for a website but did not know that this site had to be enabled with a console command as well. Last, four participants could not manage to start Apache2 after the edit of the configuration file. In every on of these scenarios, we observed that the participants did troubleshooting, e.g., by searching the web or looking at video tutorials, but based on their statements, we conclude that they did not fully understand the process and the corresponding environment.

One case was particularly noteworthy: Participant P5 who started with the CA-Certbot condition failed the Apache task, i.e., did not manage to correctly configure Apache to host the HTML files, but managed to correctly operate Certbot and completed the security configuration without task-related support. In the following CA-Traditional task, P5 managed to configure Apache but then failed to properly install the certificate. This lends further support to our finding that it is not lack of security skills or knowledge causing difficulties: The

common source of difficulty is the Apache environment.

In total, 28 participants successfully managed to execute the main task in the CA-Certbot condition, whereas 16 did so in the CA-Traditional condition. McNemar's chi-square test ($p = 0.0015$, 95% confidence interval from 1.527 to 28.563) indicates a statistically significant higher completion rate in CA-Certbot (90%) than in CA-Traditional (52%). The McNemar test was used because we were operating on paired data.

As stated before, half of the participants interacted with Certbot first, and the other half started with the traditional CA. In both cases, we saw that the success rates were slightly higher for the second condition, which could indicate a learning effect. Overall, the CA-Certbot treatment had four failures when it came first, and no failure when the task was completed as the second task. The CA-Traditional treatment had eight failures when it came first, and seven when it came second. However, the differences were not statistically significant (Fisher's exact test $p = 0.226$ and $p = 0.724$, respectively).

Number of web servers	Fail both	Success CA-Certbot only	Success CA-Trad. only	Success with both
0	2	3	0	1
1–5	1	8	0	9
≥ 6	0	1	0	6
Sum	3	12	0	16

TABLE 7.6: Success rate depending on the number of web servers the participants had configured previously.

Table 7.6 gives a more detailed within-subjects view and shows the distribution of the outcome according to the number of web servers participants reported to have configured previously[8]. As shown, no participant who managed to successfully use CA-Traditional failed at using CA-Certbot (Success in CA-T only). However, 12 participants who succeeded with CA-Certbot failed in CA-Traditional (Success in CA-C only). Four of them started with the CA-Certbottask and eight with the CA-Traditionaltask. The results suggest that the higher the number of servers a participant had configured previously, the fewer double failures occurred (no success in either condition). In the one to five servers bin, roughly half the participants (eight of 18) managed only CA-Certbot, and half managed both (nine of 18). In the six or more servers bin, almost all (six of seven) managed both. This shows that Certbot (i.e., the CA-Certbot condition) is particularly useful for less experienced administrators.

[8]The questionnaire provided the answer bins 0, 1, 2–5, 6–15 and 16+. The bins 1 and 16+ had very few respondents; thus, we combined the bins with the adjacent bins for ease of analysis.

	CA-Certbot	CA-Traditional
TLD only	8	1
WWW only	5	3
Both	15	13

TABLE 7.7: Distribution of the domains the participants chose to include in the certificate separated by the CA condition. "TLD only" and "WWW only" mean that they entered only "tld.com" or "www.tld.com" as a valid domain.

However, there was one exception in which the CA-Traditional condition did better than the CA-Certbot task. It concerned the valid domain names a certificate includes. Although not a technical specification, it is a common convention that "tld.com" points to the same website as "www.tld.com". A problem that can arise is that a certificate which is issued for only one of these domains triggers a warning for the other domain. Table 7.7 shows the domains that the participants chose for their certificate. In the CA-Traditional condition, 13 participants configured their certificates to work for both options. Only four picked only one or the other. In the CA-Certbot condition, 15 configured their certificates to be valid for both options, but 13 picked only one or the other. However, this difference was not statistically significant (McNemar test $p = 1.00$).

7.4.2 Efficiency

For the 16 participants who succeeded at both tasks, we observed the amount of time these participants needed to enable TLS on their server. The time was derived from the video analysis in combination with timestamps collected from the bash histories. We consider the time span as the interval from certificate acquisition to the end of the TLS deployment process. For the CA-Certbot task, we observed a minimum time of six minutes. The maximum was 52 minutes, with a median of 18 minutes ($Mean = 21$, $SD = 15$). For the CA-Traditional task, the participants needed at least 23 minutes, and up to 113 minutes with a median of 65 ($Mean = 57$, $SD = 27$). A comparison of the two groups, the time participants needed for the CA-Certbot task ($Median = 18$) was statistically significantly less than for the CA-Traditional task ($Median = 65$; Wilcoxon signed rank test, $V = 2$, $p < .0027$).

7.4.3 Security Analysis

After the study was finished, we analyzed all final server configurations using the "Qualys SSL Server Test" to identify the TLS-configuration properties, and thus, the resulting security. Qualys presents its user a rating for the server depending on the quality of their SSL configuration. Table 7.8 shows the outcome

		CA-Cbot	CA-Trad.
Grade	A+	2	2
	A	11	11
	A-	0	3
	B-F	0	0
	T	3	0
Key Size	2048	15	0
	4096	0	16
	EC256	1	0
Forward Secrecy	Fully	16	13
	Incomplete	0	1
	Not Available	0	2
HSTS	Yes	3	3
	No	13	13

TABLE 7.8: The security results we observed for each CA for participants who finished both tasks ($n = 16$).

for the 16 participants who finished both tasks divided into the CA-Certbot group and the CA-Traditional group. Regarding the grade, nearly all configurations got at least an A, meaning no known attacks on the protocol were exploitable, and the key size was large enough. In the CA-Certbot group, we observed three domain-name mismatches: The domain from the certificate delivered by the server did not match the domain name from the server because the participants forgot to include "www" as a prefix for the domain name. This resulted in a capped grade T (not Trusted), which otherwise would have been an A-rated configuration. The reason that CA-Certbot did worse than CA-Traditional in these cases can be traced to the documentation used. In the three failure cases, the CA-Certbot participants simply followed the instruction of the tool, which does not mention or offer the www sub-domain. Whereas the tutorials used by the CA-Traditional participants made them aware of the www sub-domain, because it was suggested in an example together with the plain domain. The participants with an A+ grade extended the automatic configuration (CA-Certbot) or the manual configuration (CA-Traditional) with additional features, such as enabling HSTS. Due to the instructions given on the CA-Traditional homepage, all participants generated a key with a key size of 4096 bits compared to the 2048-bit keys generated by Certbot that were used 15 times. One participant, however, followed instructions on some website that generated the key using elliptic curves and a key size of 256 bits. Forward Secrecy was fully enabled by all 16 participants in the CA-Certbot group and 13 in the CA-Traditional group. Only one participant enabled Forward Secrecy incompletely, and two did not manage to enable it. In both conditions, three participants enabled HSTS, while all others did not.

Comparing the results to those of Krombholz et al. [79], the participants achieved higher grades. Although most of the participants' configurations resulted in the grade B (16 of 28), and only four got an A, the participants in the present study who finished CA-Traditional ($n = 16$) were graded with at least an A- (see Table 7.8). However, the CA we used provided examples which the minimal CA of Krombholz et al. did not. However, only four participants had an invalid configuration in Krombholz et al.'s study, compared with 15 in the present study. This result can be explained by two factors, firstly the study setup contained the entire process, and thus, was more complex than the Krombholz et al's study. Second, unlike Krombholz et al., we did not filter based on skill, and therefore, had a wider range of skill sets in the participant sample.

Comparing the results to Bernhard et al. [19], the participants had more success. In Bernhard et al.'s first study zero out of nine participants managed to use the traditional CA, and only four out of nine managed with Let's Encrypt. In their second study, three out of five managed with the traditional CA, and the same number managed with Let's Encrypt. As no details were reported at which steps the participants failed, and skill was not measured with a questionnaire but self-reported, a more detailed comparison is not possible.

7.4.4 Support

To observe the usage of the Mattermost support and feedback channel, we recorded the time and the reason for which a participant contacted us. Twenty-five participants used the channel and asked 52 questions. Because the categorization of these messages was critical for all other results, we followed a two-stage coding procedure: First, three coders independently coded all support interactions using the categories. We calculated an initial Fleiss' kappa (0.5) and Krippendorff's alpha (0.5) [78]. With three coders and eight categories, values in this range are to be expected. All codes with disagreement were discussed, and full agreement was reached in the second round of coding. For coding categories, see Table 7.9.

To simplify the analysis with respect to success, we grouped participants in categories from 0 to 4 as participants who received only non-CA-specific support. They did not receive any information relevant to the success or failure of the CA conditions that they did not already have in the task description. Category 5 participants were labeled as having received "technical help" while also being counted as receiving non-CA-specific support. The distinguishing factor for technical help was that the problem had to be the same for both CA conditions, e.g., SSH or permission problems. As a counter-example, we had two participants who had problems installing Python. This was not categorized as a general technical problem, because installing Python was needed only for the CA-Certbot condition and not in the CA-Traditional condition, and thus, critical to the CA aspect. Categories 6–8 were given if questions were specific to one of the two CA conditions. Thus, participants who needed this kind of help fell into

	Category	Name	Description
Non-CA-specific support	0	No Support Contact	
	1	Self-Help	Participant solved the problem before the support experimenter had to intervene.
	2	Study Description	Questions related to information that had been handed out in the study description. The support experimenter simply repeated information from the task description.
	3	A*	Questions that went above and beyond what was expected of participants; e.g., participant asked whether we would prefer ECC over the default RSA. The support experimenter would give the answer closest to the default option.
	4	Off-Topic	Messages that had no relation to the task or useful information, e.g., "What is my study ID?"
	5	General Technical	Problems with standard Unix commands, which affect both CA conditions equally, e.g., problems with SSHing onto the study server.
CA-specific support	6	Nudge	Conversations where the support experimenter nudged the participants to think for themselves, e.g., answering a question by saying, "This is up to you."
	7	Hint	The support experimenter sent a concrete hint for how to solve a CA-related problem, for instance, a command to run the Certbot.
	8	Active Help	The support experimenter executed part of the task for the participant, e.g., generating the signed certificate because the participant was not likely to succeed within the time allotted, and we wanted to gather information on how the next steps would play out.

TABLE 7.9: Support Categories

Name	CA-Certbot			CA-Traditional			Total
	# Ques.	# Part.	Success rate	# Ques.	# Part.	Success rate	# Ques
General Technical	6	4	50%	23	10	37%	29
Study Description	3	3	100%	7	7	71%	10
Active Help	1	1	0%	5	4	0%	6
Hint	0	0	-%	4	2	0%	4
A*	0	0	-%	1	1	100%	1
Nudge	0	0	-%	1	1	0%	1
Self-Help	0	0	-%	1	1	100%	1
Not Contacted/ Off-Topic	(3)	24	92%	(1)	16	56%	0
Total questions	10			42			52

TABLE 7.10: Support overview and success rates

the CA-specific support category. Only one participant received only a single nudge; thus, category 6 did not carry much relevance for further analysis. As stated in section 7.3, interventions that fell in categories 7 and 8 were measures of last resort, and we classified the associated tasks as failed, but used the data separately to judge their relative difficulty. For more on this, see subsection 7.4.1.

Table 7.10 shows the results of the support coding in descending order. The participant count does not add up to 31, because participants can be listed in multiple categories depending on the type and number of questions that they asked (excluding off-topic questions). There were almost three times as many support requests in the CA-Traditional condition compared to the CA-Certbot condition, and there were nine times as many category 7 and 8 support interventions, which indicates that the usability of Certbot is superior. A further noteworthy indicator is that 22 of 24 (92%) participants who did not contact support managed to successfully use CA-Certbot, and only 9 of 16 (56%) successfully configured with CA-Traditional. In five cases, the experimenter actively supported the participants (category 7 or 8) because otherwise they would not have been able to complete the task. Two participants were not able to acquire a certificate under the CA-Traditional condition, and thus, received instructions for the installation part of the task. One participant failed to enable Apache2's mod_ssl plugin to enable TLS, and two others did not manage to restart the Apache2 web server due to an Apache configuration error. As stated before, we did not count these participants as succeeding in that condition.

7.4.5 User Feedback

After being exposed to a condition, participants were asked to fill out a survey concerning the task they had just completed. At the end of the survey for the second task, they were additionally asked to complete a final survey on which comparative questions were asked.

CA-Certbot Survey

After completing the CA-Certbot task, participants were asked if they had previously heard of Let's Encrypt, and to describe the purpose of the software in their own words. All answers were gathered and coded by two researchers. Fourteen (of 31) had already heard of Let's Encrypt. We identified that most of the answers mentioned that Let's Encrypt is a certificate authority (19 participants) that issues free certificates (11) to secure communication with a web server (8).

In each task survey, we asked participants which task-related steps they considered easy and which they considered hard. Of the 31 participants who finished CA-Certbot and filled out the survey, six mentioned that it was difficult to configure the Apache2 web server. For example, P2 addressed the configuration of an automatic redirect: *"Adding another host to the non-SSL redirects turned out [to be] annoying, Certbot did not completely fix the configuration files on –expand*

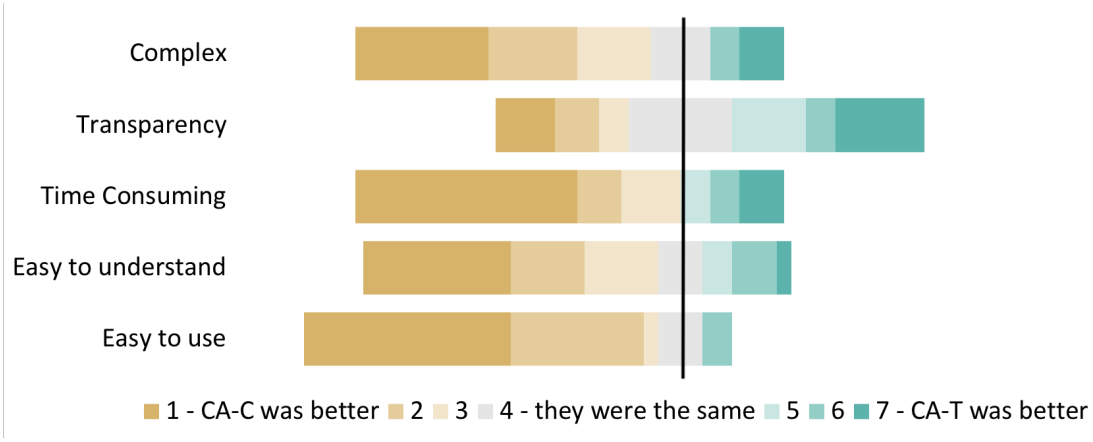


FIGURE 7.2: Participants' perceptions of the two tasks ($n = 16$, those who succeeded in both)

mode. [sic]" Two participants mentioned that the "large" amount of documentation for Let's Encrypt was hard to understand. However, one of them stated that it was "still very good" (P26). Finally, participants desired more information about what Certbot does, and wished to understand "what is happening in the background" (P28). Concerning the easy parts of the configuration process, many participants mentioned Certbot itself (12) followed by the configuration (two) and the ease of the overall process due to Certbot (two).

CA-Traditional Survey

Following the CA-Traditional task, we asked the same questions. Six participants reported problems with deploying the certificate in the Apache2 configuration, and four had difficulties understanding the documentation. P26 commented *"Each step was not very easy to understand. There should have been more details or explanations."* Concerning the tasks that were perceived as easy, seven participants mentioned the documentation because *"it basically was just copy pasting"* (P11) followed by easy key generation (three).

Comparative Survey

In the final survey, we asked the participants to compare the two tasks in terms of the five aspects: How "Easy to use," "Easy to understand," "Time-consuming," "Transparent," and "Complex" were the systems?

Figure 7.2 shows the plotted outcome of this question set for participants who completed both tasks successfully. It is based on a 7-point scale ranging from 1 (CA-Certbot was better), to 4 (they were the same), to 7 (CA-Traditional was better). In all categories except "Transparent," CA-Certbot performed better than CA-Traditional. It seems that the level of automation that Certbot offers reduced the perception of transparency.

7.5 Limitations

This study has several limitations that must be considered when interpreting the results. The sample consisted of computer science students from one institution. Although there is growing evidence that computer science students are useful proxies for these kinds of studies (Krombholz et al. [79], Yakdan et al. [143], Naiakshina et al. [97]), the results should not be over-interpreted and we caution against using the absolute numbers from this study to infer how a wider administrator population would fare. In particular, the trouble some of our participants had with file permissions is unlikely to affect seasoned administrators. However, it is likely that there are also varying skill levels among real administrators, and thus, we think that the insights gathered from the mix of skill levels is useful. We are also confident that the overall results of the A/B test are useful despite this limitation.

This study was also limited by the laboratory setting. It is likely that had the participants performed these tasks in a production environment with real-world security implications, they would have behaved differently. In a real setting, the participants could also have have taken more time.

Finally, the two separate time limits could have introduced a bias, which we did not think of beforehand. Although the two- and three-hour limits were grounded in the pre-studies, we did not consider the possible interaction between the two. It is possible that outcomes were affected due to a difference in learning and fatigue between the conditions. We discuss both possibilities and contrast this setup with a study setup with a three-hour limit for each of the two tasks.

Luckily, only two participants (P9 and P15) ran into the two-hour time limit for the CA-Certbot task. Both started with the CA-Certbot task. They also both failed the CA-Traditional task. If they had had three hours instead of two for the CA-Certbot task, they might have succeeded in the CA-Certbot task, and they might have learned enough during that additional hour to then also succeed in the CA-Traditional task. To judge the likelihood of either of these options, we analyzed the bash and web history of both participants. Both spent a lot of time getting familiar with the file and permission system, as well as the Apache2 configuration files. Even though it is possible that these two participants would have succeeded in their tasks if they had had one hour more, we do not think it is likely. To put this into context, participants who succeeded in the CA-Certbot task needed a median of 18 minutes ($Mean = 21$, $SD = 15$) to finish their tasks. Those who succeeded in the CA-Traditional task needed a median of 65 minutes ($Mean = 57$, $SD = 27$). Thus, although the different cut-off times were not a good design choice, they did not seem to have a negative impact on the results.

7.6 Recommendations

7.6.1 Recommended Improvements for Certbot

The findings presented in section 7.4 clearly show that the designers of Certbot have done an excellent job in making TLS configuration easier and faster. Certbot outperforms the traditional approach in almost all areas. In particular, the automation of the Apache-related tasks proved to be beneficial to the participants. But there is still room for improvement. The biggest negative aspect we found is that participants consistently ranked Certbot's transparency lower than the manual approach, saying things like *"everything was easy, but [...] Certbot is not transparent to me. I do not know what it actually did and the whole process inside, for me it is like (a) black box"* (P21) or *"which is a little worrying for security-related tasks in my opinion"* (P28). Although Certbot offers a verbose option, none of the participants made use of it. As we saw the main benefit in automating the Apache steps, it is an interesting avenue for future work to explore whether additional manual steps would have a negative or positive impact on the overall usability, security, and perception of the system.

In addition, the use of additional security features was not obvious to participants, and is not contained in the Certbots' default workflow: *"The problem is that, with Certbot you cannot use HPKP, OCSP Must-Staple or Expect-CT, because you don't get a fixed private-key, and no control over the CSR"* (P17). Because Certbot is the recommended command line tool for Let's Encrypt, it has to cover many use cases and different types of administrators. However, offering users more advanced security configurations per default could be beneficial for the overall security. But this path has to be trodden with care. Although some experts missed advanced settings and extended configuration possibilities, we argue that Certbot is on the right path, because it is making security usable for most users. Nevertheless, future work should look into the tradeoff between security and generalizability.

A final minor observation concerns the "www" sub-domain. As stated previously it is a common convention that "www.tld.com" leads to the same location as the plain domain "tld.com." However, this is not a requirement. Currently, Certbot expects the administrator to know about this technicality and manually specify both options. Alashwali et al.'s study [12] found, that "www" domains tend to have a stronger security than their related plain domains. In this study, we saw a similar pattern, as many participants failed to include both domains. Considering the huge scale of LE and Certbot, this can lead to an even larger number of false positive warnings than Akhawe et al. found [11]. We recommend to prompting a dialog to the user that offers the option to directly issue the certificate for both domains with an explanation why this can make sense.

7.6.2 Lessons Learned from Certbot

Most academic papers highlight usability failures when examining security solutions. We studied Certbot because the general perception was that Certbot offered good usability. The study results confirms this perception. The EFF's Certbot and Let's Encrypt offer vastly better usability, leading to significantly higher success rates in less time. Therefore, we want to take this opportunity to see whether there are lessons to be learned and applied to other application areas. In our assessment, one of the key factors of Certbot's success is its simplicity born through the good design decision of a team of experts combined with good administrator-centered engineering. Participants did not need to know much about what was going on. Certbot applied the knowledge of its experts automatically with little need for specialized knowledge, by guiding the user through the process using a dialog-like approach instead of requiring multiple commands on the command line. Looking back, it is interesting to note for how long HTTPS configuration was considered a hard problem to solve at scale. Although the concept that a small group of experts decides what is best for the community is not without risk, from a usability perspective it offers a lot of potential.

The two main components of the good usability stem from automation and safe defaults. Certbot automated seven steps while introducing only two new manual steps (see Table 7.5). Certbot also uses safe defaults for most security properties. The only bigger disadvantage of Certbot was that participants felt that it lacked transparency.

The question is whether Certbot' success can be replicated in other areas. For this, we need to look at several properties of the HTTPS scenario. First, we discovered that it was mainly the automation of the Apache steps that reduced failures. Although automating the other steps saved time and improved overall usability, what would classically be seen as the *difficult* steps, i.e., where the admin has to interact with cryptographic concepts, such as key generation and signing, actually did not lead to failures. As we discuss below this suggests that a good portion of research in the field of usable security and privacy might have focused on the less important parts. Second, the other implication from the fact that the Apache automation is that Certbot profits from the fact that it needs to support only a limited number of web servers. Third, there are clear recommendations about what are considered safe defaults, e.g., what key size is sufficient, what ciphers and protocol versions should be used, etc.

The attributes listed above do not lend themselves to all areas. Thus we present both scenarios in which we believe the Certbot approach can work, as well as some where other concepts need to be found.

eMail/Messaging

Secure email is one of the bogeymen of computer security that has been plaguing usable security researchers in the end-user realm for decades [140, 57, 112]. Although standards like PGP [9] and S/MIME [10] have been around for a long time, adoption is minimal. Potentially, one of the problems is that usable security researchers have mainly targeted end-users, and not developers and administrators. Offering a simple Let's Encrypt-like service which allows administrators of an organization to roll out free and easy-to-use certificates to users, and take the burden of publishing and finding keys from them, might turn out to be a missing link. This scenario, of course, is a much more challenging than the one Let's Encrypt currently addresses. The heterogeneous environment and the large number of different components involved increase the difficulty.

Whatsapp[11], for example, hides the whole key exchange process from its users while enabling full end-to-end-encryption. Like other centralized messaging services, the engineering needed to do this is far less than in the heterogeneous email environment. However, the high adoption rate shows the promise of automating key management for end-user messaging. Thus, taking a Certbot approach to email encryption could be worthwhile, and we would like to see the usable security community look at the administrator and developer side of this old problem.

Password Storage

Research by Naiakshina et al. [98, 97] showed that students and software development freelancers have many difficulties when trying to store passwords securely. We see several parallels to the TLS configuration scenario. In both cases, a small number of cryptographic steps need to be taken. From the point of view of security experts, these steps are fairly easy and as HTTPS, recommended safe choices are available. But many participants did not know all steps (salting, hashing and iterations), or were not up to date. For instance, many thought that MD5 was still acceptable, or used Base64 encoding to store the passwords "securely". Although many libraries offer secure storage, there is no highly visible authority and no generic approach. An initiative with a tool that can generate secure password storage code by providing a standardized and secure method by default in a number of different languages could offer similar improvements as Certbot. However, the challenge is that the number of different languages is large, and environments are more heterogeneous, which increases the technical complexity of the tool.

[9] `https://tools.ietf.org/html/rfc3156`, Accessed: 09/02/2019

[10] `https://tools.ietf.org/html/rfc1847`, Accessed: 09/02/2019

[11] `https://www.whatsapp.com/security/WhatsApp-Security-Whitepaper.pdf`, Accessed: 09/02/2019

Firewall Configuration

An area of usable security research where the Certbot approach is less likely to work as well is enterprise firewall configuration. The task itself is mostly procedural; however, important security decisions specific to the administrators' goals have to be made, which was identified as a challenging area for usable security research by Edwards et al. [40]. Administrators are often confronted with difficult decisions concerning edge cases about which packets should be discarded. Those configurations are bound to functional consequences, and giving a "one fits all" solution is hard. The functional steps, i.e. the configuration, can be supported with good usability [134]. However, the decisions that operators have to make cannot be easily automated, and other forms of usability research are needed.

Update Management

Similar to the task of firewall configuration is the case of update management for administrators who manage heterogeneous environments. Each different platform and software increases the complexity of the task and hinders simple automation. The process involves multiple stakeholders, and the decisions have consequences that impact the security and availability of systems. Previous work showed that automatic updates are not "universally suitable" for a corporate context [85]. The update process spans multiple stages, different policies and things to consider, such as disruptions in the others' workflow. While updates have dependencies on other parts, additional usable security research is needed.

7.7 Lessons Learned Concerning Administrator Study Design

We studied a complex administrative task in the lab to conduct an A/B comparison of Certbot and a traditional CA. As usable security research into administrators and developers is still a young field with little methodological experience, we would like to discuss insights gained from this extensive five-hour lab study.

7.7.1 Interaction via Support Channel

Allowing interaction between the experimenter and participants brings several risks. First, there is the risk that the experimenter fails to treat all participants equally. This can be countered to a certain extent by using a playbook (see section D.4) that defines what actions an experimenter is allowed to take, and has ready-to-use texts. Second, even if the experimenter is consistent, they might still influence the results by the playbook favoring one condition or another.

A careful and neutral design is needed to avoid this risk. Finally, the use of a support channel can influence the time participants need for a task and make the evaluation more complex, because the number of result categories is higher (*succeeded without help, succeeded with help, failed without help, and failed with help*). Despite these risks, we found the support channel offered very valuable insights into the study subject and very natural interaction. For instance, an insight we would have lost had it not been for the support channel was that one participant failed to perform the domain configuration of Apache but succeeded in using Certbot. The participant also failed to configure the traditional CA. Without the support channel, it would have looked like the participant had failed at both approaches. However, with the interaction, we saw that Certbot's usability is so good that even someone who struggles with simple configuration tasks can use it. We also gathered interesting comments and feedback from the chat. On the whole, we think the benefits outweigh the drawbacks.

7.7.2 Framing

We used two different study descriptions. One was a very simple description that made no attempt at realism or hiding the fact that it was a study task. The second introduced a role-playing scenario in an attempt to be more realistic. It used custom domains, websites, and user credentials to facilitate the role-playing scenario. We did not see any difference in behavior based on these two different frames, and thus, the substantial extra effort needed to create a more realistic study setting when designing studies for administrators in the lab context does not seem necessary. However, the lab study setup itself could have framed the participants in such a way that the scenario description did not have an influence on the outcome and that other mechanisms, e.g., field studies where the participants deploy a certificate for their own site, have to be researched. We found indicators that nudging people to security results in better security outcomes. More work is needed to analyze the influence of these factors.

7.7.3 Measuring Performance

The duration and degrees of freedom from the participants' perspective have an impact on the broad range of possible outcomes. In this study design, the participants had the possibility of choosing a non-linear way of solving the task. We used a time-consuming approach and manually tracked all user actions by watching the recorded sessions. But even then it was not easy to decide when a certain task was stopped, another one started, or a previous one was resumed. It also was hard to tell if a participant was taking a break. Requesting participants to log this would have led to an increased mental load for them, and thus, reduced the focus and created a more artificial situation. Automated approaches for this kind of task tracking would be extremely useful.

7.7.4 Expertise and Study Design

As mentioned in subsection 7.3.3, unlike Krombholz et al., we invited all students who completed the pre-screening survey to participate in the lab study, independent of their pre-screening score. The rationale for excluding low-scoring participants is to conserve study resources. There is little value in having a participant who lacks basic skills take part in an administrator study. Although it is less critical for a within-subjects design, unfit participants could seriously skew between-subjects studies. However, taking only the best participants, as in the Krombholz et al. study, skews the results as well. It would be ideal to have a pre-screening survey with which to filter participants who lack the basic skills without also losing low-skilled participants. Unfortunately, our showed that most of the screening questions were not good predictors of participants' performance. In this study only the number of previously configured servers seemed like a promising predictor.

We saw a similar picture in a developer study conducted by Wermke et al., who found a correlation between years of programming experience and success in the tasks [5]. However, a similar study by Naiakshina et al. [98] failed to find the same correlation.

Thus, although expertise is undoubtedly important for the outcome of expert studies, assessing expertise is very hard. The difficult pre-screening process makes between-subjects study designs particularly risky, and we recommend using within-subjects designs whenever possible. At the same time, we encourage more work on assessing skill levels using questionnaires, to enable reliable balancing in future work.

7.8 Ethical Considerations

All participants signed a consent form with a description of the tasks and information about data collection. They were informed about the screen-recording software and the collection of their browser and bash histories. Participants were also told that we would not rate any of their solutions, and that we were interested only in the process of how they executed their tasks, to prevent an exam-like situation, which could make them feel uncomfortable or under pressure, and introduce some kind of desirability bias. The consent form, as well as the study, was approved by our university's IRB. All collected data was processed and stored in compliance with the strict general data protection regulation (GDPR) of the European Union.

7.9 Summary

In this chapter, I conducted a randomized control trial to compare the usability of two different approaches of configuring HTTPS for an Apache web server.

This study compared the EFF's Certbot, the recommended command line tool for Let's Encrypt CA, with a traditional approach that uses Let's Encrypt in the back-end. I showed that the EFF's Certbot is significantly easier and faster to use for all participants' skill levels. As a consequence of such improved usability aspects, significantly more users were able to set up a secure HTTPS configuration using LE than using the traditional approach. I identified that automation of steps pertaining to the configuration of Apache drove the increased success rate. Key generation, signing, and other cryptographic and CA-related steps did not cause the problems that might have been assumed.

Chapter 8

Conclusions

In their daily work, administrators have to deal with many, sometimes security-related, tasks, while at the same time having to follow functionality requirements. My thesis shows that if we find ways to support them to act securely, e.g., by providing solutions with secure defaults, this significantly improves the security of a large number of systems. Therefore, understanding administrators' tasks, their environment, and their problems need to be one of the essential areas in future usable security research that is just beginning to emerge.

In this thesis, I researched two tasks that play a significant role in administrators' work and have a high impact on IT security in the corresponding fields: updates and TLS configuration.

First, I presented a mixed-methods study that revealed how administrators deal with security updates in their working context, what obstacles they are facing, and where they get information about updates. Out of this work, I created a model that split the process into six different stages. The results imply that even for experienced administrators, the consequences of applying updates are hard to predict, and one driving factor in delaying updates are downtimes. Another observation was that administrators often rely on information provided by third parties instead of the vendor and consult online sources in the consideration to update. This work's findings motivated the two studies I presented in chapter 4 and chapter 5.

In the following case-study, with its goal to further learn about the update process, I presented another mixed-method study. By conducting interviews, a survey, and analyzing the ticket system in a web development company, I showed that the process identified in the previous study was not flexible enough to match the company's observed processes. After presenting examples that explain this problem, I developed a more flexible model that added additional elements like external interrupts that the first model missed.

To find more information about the information sources and the relevance of specific details they contain, I presented further interviews and a survey with administrators. This study showed that while attaining information, the key information for system administrators consists of the purpose, dependencies, and known issues of an update.

Out of this research, several topics emerge that motivate future work. One can investigate current established formal processes and evaluate their effectiveness in supporting timely updates on a larger scale than the presented case study. Computer-supported solutions could be researched further that enable better communication between administrators and, in this way, enhance the transfer of knowledge, like Jenkins et al. [72] already started to investigate. Also, feasible tools that support situational awareness should be developed and researched, e.g., by helping administrators find out about relevant updates and provide them with the information they need.

In the last part of this work, I presented a lab study comparing the usability of two different approaches to configuring HTTPS for an Apache webserver. It showed that the automated approach, using EFF's Certbot, is significantly simpler and faster to use for all participants' skill levels compared to the manual approach. This work highlights a case where a tool improves both usability and security. Its principles can be used as a blueprint to inform further research like automated password encryption in databases or a better setup procedure in email encryption.

This thesis aimed to extend the research field of Usable Security and Privacy to understand how administrators update software and systems in a corporate context and how automation influences the TLS configuration process. The methodology developed as part of this thesis can be used as a basis for further studies into administrator behavior since the four studies only cover a small excerpt of the various tasks that administrators have to execute and are responsible for. This can be securing company networks, managing password policies, or adapting to the emerging use of Internet-of-Things devices in a corporate context, just to name a few. This makes researching administrators a pivotal and promising field for usable security research because so little work has been done in this area, and even small improvements can have an enormous impact.

Bibliography

[1] Surafel Lemma Abebe, Nasir Ali, and Ahmed E. Hassan. "An Empirical Study of Software Release Notes". In: *Empirical Softw. Engg.* 21.3 (June 2016), 1107–1142. ISSN: 1382-3256. DOI: 10.1007/s10664-015-9377-5.

[2] Y. Acar et al. "Comparing the Usability of Cryptographic APIs". In: *2017 IEEE Symposium on Security and Privacy (SP)*. 2017, pp. 154–171. DOI: 10.1109/SP.2017.52.

[3] Yasemin Acar, Sascha Fahl, and Michelle L Mazurek. "You are not your developer, either: A research agenda for usable security and privacy research beyond end users". In: *Cybersecurity Development (SecDev), IEEE*. IEEE. 2016, pp. 3–8.

[4] Yasemin Acar et al. "Developers Need Support, Too: A Survey of Security Advice for Software Developers". In: *Cybersecurity Development (SecDev), 2017 IEEE*. IEEE. 2017, pp. 22–26.

[5] Yasemin Acar et al. "Security Developer Studies with GitHub Users: Exploring a Convenience Sample". In: *Symposium on Usable Privacy and Security (SOUPS)*. 2017.

[6] Yasemin Acar et al. "You Get Where You're Looking for: The Impact of Information Sources on Code Security". In: *Proceedings - 2016 IEEE Symposium on Security and Privacy, SP 2016* (2016), pp. 289–305. DOI: 10.1109/SP.2016.25.

[7] Anne Adams and Martina Angela Sasse. "Users are not the enemy". In: *Communications of the ACM* 42.12 (1999), pp. 40–46.

[8] David Adrian et al. "Imperfect Forward Secrecy: How Diffie-Hellman Fails in Practice". In: *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security*. CCS '15. Denver, Colorado, USA: ACM, 2015, pp. 5–17. ISBN: 978-1-4503-3832-5. DOI: 10.1145/2810103.2813707.

[9] Maarten Aertsen. *How to bring HTTPS to the masses? Measuring issuance in the first year of Let's Encrypt*. https://www.sidnlabs.nl/downloads/theses/How-to-bring-HTTPS-to-the-masses_measuring-1y-of-LE.pdf. [Online; accessed Februar 2019]. 2016.

[10] Maarten Aertsen et al. "No domain left behind: is Let's Encrypt democratizing encryption?" In: *Proceedings of the Applied Networking Research Workshop*. ACM. 2017, pp. 48–54.

[11] Devdatta Akhawe et al. “Here’s my cert, so trust me, maybe? Understanding TLS errors on the web”. In: *WWW 2013 - Proceedings of the 22nd International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2013, pp. 59–69. ISBN: 9781450320351. DOI: 10.1145/2488388.2488395.

[12] Eman Salem Alashwali, Pawel Szalachowski, and Andrew Martin. “Does "www." Mean Better Transport Layer Security?” In: *Cryptology ePrint Archive, Report 2019/941*. https://eprint.iacr.org/2019/941. 2019.

[13] Johanna Amann et al. “Mission accomplished?: HTTPS security after diginotar”. In: *Proceedings of the 2017 Internet Measurement Conference*. ACM. 2017, pp. 325–340.

[14] Hala Assal and Sonia Chiasson. “Security in the Software Development Lifecycle”. In: *Fourteenth Symposium on Usable Privacy and Security (SOUPS 2018)*. USENIX Association. 2018, pp. 281–296.

[15] Nimrod Aviram et al. “DROWN: Breaking TLS Using SSLv2.” In: *USENIX Security Symposium*. 2016, pp. 689–706.

[16] Rekha Bachwani et al. “Mojave: A Recommendation System for Software Upgrades.” In: *MAD*. 2012.

[17] Rob Barrett et al. “Field studies of computer system administrators”. In: *Proceedings of the 2004 ACM conference on Computer supported cooperative work - CSCW ’04*. New York, New York, USA: ACM Press, 2004, p. 388. ISBN: 1581138105. DOI: 10.1145/1031607.1031672.

[18] Ofer Bergman and Steve Whittaker. “The Cognitive Costs of Upgrades”. In: *Interacting with Computers* 30.1 (2017), pp. 46–52.

[19] Matthew Bernhard et al. “On the Usability of HTTPS Deployment”. In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI ’19. Glasgow, Scotland Uk: ACM, 2019, 310:1–310:10. ISBN: 978-1-4503-5970-2. DOI: 10.1145/3290605.3300540.

[20] John M. Blythe and Lynne Coventry. “Costly but effective: Comparing the factors that influence employee anti-malware behaviours”. In: *Computers in Human Behavior* 87 (2018), pp. 87 –97. ISSN: 0747-5632. DOI: https://doi.org/10.1016/j.chb.2018.05.023.

[21] John M Blythe, Lynne M Coventry, and Linda Little. “Unpacking Security Policy Compliance: The Motivators and Barriers of Employees’ Security Behaviors.” In: *SOUPS*. 2015, pp. 103–122.

[22] David Botta et al. “Towards understanding IT security professionals and their tools”. In: *Proceedings of the 3rd symposium on Usable privacy and security*. ACM. 2007, pp. 100–111.

[23] Robert L. Brennan and Dale J. Prediger. "Coefficient Kappa: Some Uses, Misuses, and Alternatives". In: *Educational and Psychological Measurement* 41.3 (1981), pp. 687–699. DOI: 10.1177/001316448104100307.

[24] William J. Buchanan, Scott Helme, and Alan Woodward. "Analysis of the adoption of security headers in HTTP". In: *IET Information Security* (2017).

[25] Karoline Busse, Julia Schäfer, and Matthew Smith. "Replication: No One Can Hack My Mind Revisiting a Study on Expert and Non-Expert Security Practices and Advice". In: *Fifteenth Symposium on Usable Privacy and Security ({SOUPS} 2019)*. 2019.

[26] Jeremy Clark and Paul C. van Oorschot. "SoK: SSL and HTTPS: Revisiting past challenges and evaluating certificate trust model enhancements". In: *Security and Privacy (SP), 2013 IEEE Symposium on*. IEEE. 2013, pp. 511–525.

[27] *Common Vulnerability Scoring System version 3.1*. https://www.first.org/cvss/specification-document. [Online; accessed October 2020].

[28] The commission of the european communities. *Commission recommendation of 6 May 2003 concerning the definition of micro, small and medium-sized enterprises*. 2003. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32003H0361&from=EN.

[29] *Critical Vulnerability In Profile Builder Plugin Allowed Site Takeover*. https://www.wordfence.com/blog/2020/02/critical-vulnerability-in-profile-builder-plugin-allowed-site-takeover/. [Online; accessed October 2020].

[30] *CVE Details*. https://www.cvedetails.com/. [Online; accessed October 2020].

[31] T. Dierks and E. Rescorla. *The Transport Layer Security (TLS) Protocol Version 1.2*. RFC 5246 (Proposed Standard). Updated by RFCs 5746, 5878, 6176. Internet Engineering Task Force, 2008. URL: http://www.ietf.org/rfc/rfc5246.txt.

[32] Constanze Dietrich et al. "Investigating System Operators' Perspective on Security Misconfigurations". In: *Conference on Computer and Communications Security (CCS'18)*. 2018.

[33] Diane Dodd-McCue and Alexander Tartaglia. "Self-report Response Bias: Learning How to Live with its Diagnosis in Chaplaincy Research". In: *Chaplaincy Today* 26.1 (2010), pp. 2–8. DOI: 10.1080/10999183.2010.10767394.

[34] Thomas Duebendorfer and Stefan Frei. "Why silent updates boost security". In: *TIK, ETH Zurich, Tech. Rep* 302 (2009).

[35] Zakir Durumeric, Eric Wustrow, and J Alex Halderman. "ZMap: Fast Internet-wide Scanning and Its Security Applications." In: *USENIX Security Symposium*. Vol. 8. 2013, pp. 47–53.

[36] Zakir Durumeric et al. "A Search Engine Backed by Internet-Wide Scanning". In: *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security*. CCS '15. Denver, Colorado, USA: ACM, 2015, pp. 542–553. ISBN: 978-1-4503-3832-5. DOI: 10.1145/2810103.2813703.

[37] Zakir Durumeric et al. "Analysis of the HTTPS Certificate Ecosystem". In: *Proceedings of the 2013 Conference on Internet Measurement Conference*. IMC '13. Barcelona, Spain: ACM, 2013, pp. 291–304. ISBN: 978-1-4503-1953-9. DOI: 10.1145/2504730.2504755.

[38] Zakir Durumeric et al. "The Matter of Heartbleed". In: *Proceedings of the 2014 Conference on Internet Measurement Conference*. IMC '14. Vancouver, BC, Canada: ACM, 2014, pp. 475–488. ISBN: 978-1-4503-3213-2. DOI: 10.1145/2663716.2663755.

[39] W. Keith Edwards, Erika Shehan Poole, and Jennifer Stoll. "Security automation considered harmful?" In: *Proceedings of the 2007 Workshop on New Security Paradigms - NSPW '07*. New York, New York, USA: ACM Press, 2008, p. 33. ISBN: 9781605580807. DOI: 10.1145/1600176.1600182.

[40] W. Keith Edwards, Erika Shehan Poole, and Jennifer Stoll. "Security Automation Considered Harmful?" In: *Proceedings of the 2007 Workshop on New Security Paradigms*. NSPW '07. New Hampshire: ACM, 2008, pp. 33–42. ISBN: 978-1-60558-080-7. DOI: 10.1145/1600176.1600182.

[41] EFF. *Certbot - About*. https://certbot.eff.org/about/. [Online; accessed Februar 2019].

[42] Michael Fagan and Mohammad Maifi Hasan Khan. "Why do they do what they do?: A study of what motivates users to (not) follow computer security advice". In: *Twelfth Symposium on Usable Privacy and Security ({SOUPS} 2016)*. 2016, pp. 59–75.

[43] Michael Fagan, Mohammad Maifi Hasan Khan, and Ross Buck. "A study of users' experiences and beliefs about software update messages". In: *Computers in Human Behavior* 51 (2015), pp. 504 –519. ISSN: 0747-5632. DOI: https://doi.org/10.1016/j.chb.2015.04.075.

[44] Michael Fagan, Mohammad Maifi Hasan Khan, and Nhan Nguyen. "How does this message make you feel? A study of user perspectives on software update/warning message design". In: *Human-centric Computing and Information Sciences* 5.1 (Dec. 2015), p. 36. ISSN: 2192-1962. DOI: 10.1186/s13673-015-0053-y.

[45] Sascha Fahl et al. "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. Kyoto, Japan: ACM, 2014, pp. 507–512. ISBN: 978-1-4503-2800-5. DOI: 10.1145/2590296.2590341.

[46] Sascha Fahl et al. "Why Eve and Mallory love Android: An analysis of Android SSL (in) security". In: *Proceedings of the 2012 ACM conference on Computer and communications security*. 2012, pp. 50–61.

[47] Thomas Faist. "The volume and dynamics of international migration and transnational social spaces". In: (2000).

[48] Adrienne Porter Felt et al. "Experimenting at Scale with Google Chrome's SSL Warning". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '14. Toronto, Ontario, Canada: ACM, 2014, pp. 2667–2670. ISBN: 978-1-4503-2473-1. DOI: 10.1145/2556288.2557292.

[49] Adrienne Porter Felt et al. "Improving SSL Warnings: Comprehension and Adherence". In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. CHI '15. Seoul, Republic of Korea: ACM, 2015, pp. 2893–2902. ISBN: 978-1-4503-3145-6. DOI: 10.1145/2702123.2702442.

[50] F. Fischer et al. "Stack Overflow Considered Harmful? The Impact of Copy amp;Paste on Android Application Security". In: *2017 IEEE Symposium on Security and Privacy (SP)*. May 2017, pp. 121–136. DOI: 10.1109/SP.2017.31.

[51] Marvin Fleischmann et al. "The role of software updates in information systems continuance â An experimental study from a user perspective". In: *Decision Support Systems* 83 (2016), pp. 83 –96. ISSN: 0167-9236. DOI: https://doi.org/10.1016/j.dss.2015.12.010.

[52] Alain Forget et al. "Do or do not, there is no try: user engagement may not improve security outcomes". In: *Twelfth Symposium on Usable Privacy and Security (SOUPS 2016)*. 2016, pp. 97–111.

[53] Jill J. Francis et al. "What is an adequate sample size? Operationalising data saturation for theory-based interview studies". In: *Psychology and Health* 25.10 (2010), pp. 1229–1245. ISSN: 08870446. DOI: 10.1080/08870440903194015. arXiv: arXiv:1011.1669v3.

[54] Alisa Frik et al. "Better Late(r) than Never: Increasing Cyber-Security Compliance by Reducing Present Bias". In: *Workshop on the Economics of Information Security (WEIS)*. Insbruck, Austria, 2018, p. 20.

[55] Steve Furnell. "Vulnerability management: not a patch on where we should be?" In: *Network Security* 2016.4 (2016), pp. 5 –9. ISSN: 1353-4858. DOI: https://doi.org/10.1016/S1353-4858(16)30036-8.

[56] Jonathan Gallagher, Robin Gonzalez, and Michael E Locasto. "Verifying security patches". In: *Proceedings of the 2014 international workshop on privacy & security in programming*. ACM. 2014, pp. 11–18.

[57] Simson L. Garfinkel and Robert C. Miller. "Johnny 2: A User Test of Key Continuity Management with S/MIME and Outlook Express". In: *Proceedings of the 2005 symposium on Usable privacy and security* 6 (Jan. 2005), pp. 13–24. ISSN: 1595931783. DOI: 10.1145/1073001.1073003.

[58] John R. Goodall, Wayne G. Lutters, and Anita Komlodi. "I Know My Network: Collaboration and Expertise in Intrusion Detection". In: *Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work*. CSCW '04. Chicago, Illinois, USA: ACM, 2004, pp. 342–345. ISBN: 1-58113-810-5. DOI: 10.1145/1031607.1031663.

[59] Matthew Green and Matthew Smith. "Developers are not the enemy!: The need for usable security apis". In: *IEEE Security & Privacy* 14.5 (2016), pp. 40–46.

[60] Josef Gustafsson et al. "A First Look at the CT Landscape: Certificate Transparency Logs in Practice". In: *Passive and Active Measurement*. Ed. by d Ali Kaafar, Steve Uhlig, and Johanna Amann. Cham: Springer International Publishing, 2017, pp. 87–99. ISBN: 978-3-319-54328-4.

[61] Eben M. Haber and Eser Kandogan. "Security administrators: A breed apart". In: *Soups USM* (2007).

[62] Julie M Haney and Wayne G Lutters. ""It's Scary... It's Confusing... It's Dull": How Cybersecurity Advocates Overcome Negative Perceptions of Security". In: *Fourteenth Symposium on Usable Privacy and Security (SOUPS 2018)*. USENIX Association. 2018.

[63] Marian Harbach et al. "Sorry, I Don't Get It: An Analysis of Warning Message Texts". In: *Financial Cryptography and Data Security*. Ed. by Andrew A. Adams, Michael Brenner, and Matthew Smith. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 94–111. ISBN: 978-3-642-41320-9.

[64] Norbert Hedderich. "Three Approaches to Qualitative Content Analysis". In: *Global Business Languages: Vol. 2 , Article 14* 2 (2010), pp. 162–172. URL: https://docs.lib.purdue.edu/cgi/viewcontent.cgi?article=1028.

[65] Michael Hicks and Scott Nettles. "Dynamic Software Updating". In: *ACM Trans. Program. Lang. Syst.* 27.6 (Nov. 2005), pp. 1049–1096. ISSN: 0164-0925. DOI: 10.1145/1108970.1108971.

[66] Ralph Holz, Yaron Sheffer, and Peter Saint-Andre. *Summarizing Known Attacks on Transport Layer Security (TLS) and Datagram TLS (DTLS)*. https://tools.ietf.org/html/rfc7457. [Online; accessed Februar 2019]. 2015.

[67] Ralph Holz et al. “TLS in the wild: An Internet-wide analysis of TLS-based protocols for electronic communication”. In: *arXiv preprint arXiv: 1511.00341* (2015).

[68] *How the Equifax hack happened, and what still needs to be done*. https://www.cnet.com/news/equifaxs-hack-one-year-later-a-look-back-at-how-it-happened-and-whats-changed/. [Online; accessed October 2020].

[69] *How to Manually Upgrade WordPress, Themes & Plugins*. https://www.wordfence.com/learn/how-to-manually-upgrade-wordpress-themes-and-plugins/. [Online; accessed October 2020].

[70] Dennis G. Hrebec and Michael Stiber. “A survey of system administrator mental models and situation awareness”. In: *Proceedings of the 2001 ACM SIGCPR conference on Computer personnel research - SIGCPR '01* (2001), pp. 166–172. DOI: 10.1145/371209.371231.

[71] Iulia Ion, Rob Reeder, and Sunny Consolvo. ““... No one Can Hack My Mind”: Comparing Expert and Non-Expert Security Practices.” In: *SOUPS*. Vol. 15. 2015, pp. 1–20.

[72] Adam Jenkins et al. “"Anyone Else Seeing this Error?": Community, System Administrators, and Patch Information"”. English. In: 5 (Feb. 2020). 5th IEEE European Symposium on Security and Privacy, EuroSP 2020.

[73] *Joomla! Homepage*. https://www.joomla.org/. [Online; accessed October 2020].

[74] Eser Kandogan and Eben M Haber. “Security administration tools and practices”. In: ().

[75] Moazzam Khan, Zehui Bi, and John A. Copeland. “Software updates as a security metric: Passive identification of update trends and effect on machine infection”. In: *MILCOM 2012 - 2012 IEEE Military Communications Conference*. IEEE, Oct. 2012, pp. 1–6. ISBN: 978-1-4673-1731-3. DOI: 10.1109/MILCOM.2012.6415869.

[76] Saranga Komanduri et al. “Of Passwords and People: Measuring the Effect of Password-composition Policies”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '11. Vancouver, BC, Canada: ACM, 2011, pp. 2595–2604. ISBN: 978-1-4503-0228-9. DOI: 10.1145/1978942.1979321.

[77] Michael Kranch and Joseph Bonneau. “Upgrading HTTPS in mid-air: An empirical study of strict transport security and key pinning.” In: *NDSS*. 2015.

[78] Klaus Krippendorff. “Reliability in content analysis: Some common misconceptions and recommendations”. In: *Human Communication Research* 30.3 (June 2004), pp. 411–433. ISSN: 03603989. DOI: 10.1093/hcr/30.3.411.

[79] Katharina Krombholz et al. “"I Have No Idea What I’m Doing" - On the Usability of Deploying HTTPS”. In: *26th USENIX Security Symposium, USENIX Security 2017*. 2017.

[80] Katharina Krombholz et al. ““If HTTPS Were Secure, I Wouldn’t Need 2FA”-End User and Administrator Mental Models of HTTPS”. In: *To appear in the IEEE Symposium on Security & Privacy, May 2019* (2019).

[81] Mika Latimer. *Trace-weighted binary comparison for software update management*. 2017. URL: https://www.ideals.illinois.edu/bitstream/handle/2142/99389/LATIMER-THESIS-2017.pdf.

[82] Peter Leo Gorski et al. “Developers Deserve Security Warnings, Too On the Effect of Integrated Security Advice on Cryptographic API Misuse”. In: (2018).

[83] Let’s Encrypt. *Let’s Encrypt Growth*. https://letsencrypt.org/stats/. [Online; accessed February 2019]. 2019.

[84] Frank Li et al. “Keepers of the Machines: Examining How System Administrators Manage Software Updates”. In: *Proceedings of the Fifteenth USENIX Conference on Usable Privacy and Security*. SOUPS’19. Santa Clara, CA, USA: USENIX Association, 2019, pp. 273–288. ISBN: 978-1-939133-05-2.

[85] Frank Li et al. “Keepers of the Machines: Examining How System Administrators Manage Software Updates For Multiple Machines”. In: *Fifteenth Symposium on Usable Privacy and Security (SOUPS 2019)*. Santa Clara, CA: USENIX Association, Aug. 2019.

[86] PONEMON INSTITUTE LLC. *COSTS AND CONSEQUENCES OF GAPS IN VULNERABILITY RESPONSE*. 2019. URL: https://www.servicenow.com/content/dam/servicenow-assets/public/en-us/doc-type/resource-center/analyst-report/ponemon-state-of-vulnerability-response.pdf.

[87] Antonis Manousis et al. “Shedding light on the adoption of let’s encrypt”. In: *arXiv preprint arXiv:1611.00469* (2016).

[88] Geraldine Vache Marconato, Vincent Nicomette, and Mohamed Kaâniche. “Security-related vulnerability life cycle analysis”. In: *7th International Conference on Risk and Security of Internet and Systems (CRiSIS-2012)*. IEEE Computer Society. 2012, pp. 1–8.

[89] Florin Martius and Christian Tiefenau. “What does this Update do to my Systems?–An Analysis of the Importance of Update-Related Information to System Administrators”. In: ().

[90] Arunesh Mathur et al. “Quantifying Users’ Beliefs about Software Updates”. In: *CoRR* abs/1805.04594 (2018). arXiv: 1805.04594.

[91] Arunesh Mathur et al. ""They Keep Coming Back Like Zombies": Improving Software Updating Interfaces." In: *SOUPS*. 2016, pp. 43–58.

[92] Michael Meike, Johannes Sametinger, and Andreas Wiesauer. "Security in Open Source Web Content Management Systems". In: *IEEE Security Privacy* 7.4 (July 2009). Conference Name: IEEE Security Privacy, pp. 44–51. ISSN: 1558-4046. DOI: 10.1109/MSP.2009.104.

[93] Huoy Min Khoo and Daniel Robey. "Deciding to upgrade packaged software: a comparative case study of motives, contingencies and dependencies". In: *European Journal of Information Systems* 16.5 (2007), pp. 555–567.

[94] Andreas Möller et al. "Update behavior in app markets and security implications: A case study in google play". In: *Research in the Large, LARGE 3.0: 21/09/2012-21/09/2012*. 2012, pp. 3–6.

[95] Laura Moreno et al. "Automatic Generation of Release Notes". In: *Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering*. FSE 2014. Hong Kong, China: Association for Computing Machinery, 2014, 484–495. DOI: 10.1145/2635868.2635870.

[96] Alena Naiakshina et al. "Deception Task Design in Developer Password Studies: Exploring a Student Sample". In: *Fourteenth Symposium on Usable Privacy and Security, SOUPS 2018, Baltimore, MD, USA, August 12-14, 2018*. 2018, pp. 297–313.

[97] Alena Naiakshina et al. ""If you want, I can store the encrypted password." A Password-Storage Field Study with Freelance Developers". In: *Proceedings of the 2019 ACM SIGCHI (to appear)*. 2019.

[98] Alena Naiakshina et al. "Why Do Developers Get Password Storage Wrong?: A Qualitative Usability Study". In: *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. CCS '17. Dallas, Texas, USA: ACM, 2017, pp. 311–328. ISBN: 978-1-4503-4946-8. DOI: 10.1145/3133956.3134082.

[99] James Nicholson, Lynne Coventry, and Pamela Briggs. "Introducing the Cybersurvival Task: Assessing and Addressing Staff Beliefs about Effective Cyber Protection". In: *Fourteenth Symposium on Usable Privacy and Security (SOUPS 2018)*. USENIX Association. 2018, pp. 443–457.

[100] Jon Oberheide, Evan Cooke, and Farnam Jahanian. "If It Ain't Broke, Don't Fix It: Challenges and New Directions for Inferring the Impact of Software Patches." In: *HotOS*. 2009.

[101] Marten Oltrogge et al. "To Pin or Not to Pin—Helping App Developers Bullet Proof Their {TLS} Connections". In: *24th {USENIX} Security Symposium ({USENIX} Security 15)*. 2015, pp. 239–254.

[102] Gustaf Ouvrier et al. "Characterizing the HTTPS trust landscape: a passive view from the edge". In: *IEEE Communications Magazine* 55.7 (2017), pp. 36–42.

[103] Srivatsan Parthesarathy et al. *Software update notification*. US Patent 6353926. 2002.

[104] Heike Pethe. *Internationale Migration hoch qualifizierter Arbeitskraefte*. DUV.

[105] Rahul Potharaju, Mizanur Rahman, and Bogdan Carbunar. "A Longitudinal Study of Google Play". In: *IEEE Transactions on computational social systems* 4.3 (2017), pp. 135–149.

[106] Elissa M Redmiles, Sean Kross, and Michelle L Mazurek. "How i learned to be secure: a census-representative survey of security advice sources and behavior". In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM. 2016, pp. 666–677.

[107] Elissa M Redmiles, Amelia R Malone, and Michelle L Mazurek. "I Think They're Trying to Tell Me Something: Advice Sources and Selection for Digital Security". In: *Security and Privacy (SP), 2016 IEEE Symposium on*. IEEE. 2016, pp. 272–288.

[108] Elissa M. Redmiles et al. "Asking for a Friend: Evaluating Response Biases in Security User Studies". In: *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. CCS '18. Toronto, Canada: Association for Computing Machinery, 2018, 1238–1255. ISBN: 9781450356930. DOI: 10.1145/3243734.3243740.

[109] R. W. Reeder, I. Ion, and S. Consolvo. "152 Simple Steps to Stay Safe Online: Security Advice for Non-Tech-Savvy Users". In: *IEEE Security & Privacy* 15.5 (2017), pp. 55–64. ISSN: 1540-7993. DOI: 10.1109/MSP.2017.3681050.

[110] Robert Reeder, Iulia Ion, and Sunny Consolvo. "152 Simple Steps to Stay Safe Online: Security Advice for Non-Tech-Savvy Users". In: *IEEE Security & Privacy* (2017).

[111] Robert W. Reeder et al. "An Experience Sampling Study of User Reactions to Browser Warnings in the Field". In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI '18. Montreal QC, Canada: ACM, 2018, 512:1–512:13. ISBN: 978-1-4503-5620-6. DOI: 10.1145/3173574.3174086.

[112] Scott Ruoti et al. "Why Johnny Still, Still Can't Encrypt: Evaluating the Usability of a Modern PGP Client". In: *CoRR* abs/1510.08555 (2015). arXiv: 1510.08555.

[113] Nayanamana Samarasinghe and Mohammad Mannan. "Short Paper: TLS Ecosystems in Networked Devices vs. Web Servers". In: *Financial Cryptography and Data Security*. Ed. by Aggelos Kiayias. Cham: Springer International Publishing, 2017, pp. 533–541. ISBN: 978-3-319-70972-7.

[114] Armin Sarabi et al. "Patch Me If You Can: A Study on the Effects of Individual User Behavior on the End-Host Vulnerability State". In: *International Conference on Passive and Active Network Measurement*. Springer. 2017, pp. 113–125.

[115] *Security Study: Content Management Systeme*. https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Studien/CMS/Studie_CMS.pdf. [Online; accessed October 2020]. 2013.

[116] *Shodan Heartbleed Report (11-07-2019)*. https://www.shodan.io/report/0Wew7Zq7. Online; accessed October 2020.

[117] Andreas Sotirakopoulos, Kirstie Hawkey, and Konstantin Beznosov. "On the Challenges in Usable Security Lab Studies: Lessons Learned from Replicating a Study on SSL Warnings". In: *Proceedings of the Seventh Symposium on Usable Privacy and Security*. SOUPS '11. Pittsburgh, Pennsylvania: ACM, 2011, 3:1–3:18. ISBN: 978-1-4503-0911-0. DOI: 10.1145/2078827.2078831.

[118] A.O. Stuart Schechter, R Dhamija, and I Fischer. "The Emperor's new security indicators: An evaluation of website authentication and the effect of role playing on usability studies". In: *S&P* (Jan. 2007), pp. 51–65.

[119] Joshua Sunshine et al. "Crying Wolf: An Empirical Study of SSL Warning Effectiveness." In: *USENIX security symposium*. 2009, pp. 399–416.

[120] Symantec. "Internet Security Threat Report". In: *Network Security* 21 (2016).

[121] *The best CMS and their strenghts*. https://www.ithelps-digital.com/de/blog/webseiten/cms-systeme. [Online; accessed October 2020].

[122] Yuan Tian et al. "Study on user's attitude and behavior towards android application update notification". In: *Usenix, Menlo Park, CA* (2014).

[123] Yuan Tian et al. "Supporting privacy-conscious app update decisions with user reviews". In: *Proceedings of the 5th Annual ACM CCS Workshop on Security and Privacy in Smartphones and Mobile Devices*. ACM. 2015, pp. 51–61.

[124] Christian Tiefenau et al. "A Usability Evaluation of Let's Encrypt and Certbot: Usable Security Done Right". In: *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. CCS '19. London, United Kingdom: ACM, 2019, pp. 1971–1988. ISBN: 978-1-4503-6747-9. DOI: 10.1145/3319535.3363220.

[125] Christian Tiefenau et al. "Security, Availability, and Multiple Information Sources: Exploring Update Behavior of System Administrators". In: *Sixteenth Symposium on Usable Privacy and Security (SOUPS 2020)*. USENIX Association, Aug. 2020, pp. 239–258. ISBN: 978-1-939133-16-8.

[126] *TYPO3 Homepage*. https://typo3.org/. [Online; accessed October 2020].

[127] *TYPO3 Release Notes*. https://get.typo3.org/release-notes/. [Online; accessed October 2020].

[128] Martin Ukrop et al. "Will You Trust This TLS Certificate? Perceptions of People Working in IT". In: *35rd Annual Computer Security Applications Conference (ACSAC'2019)*. ACM, 2019. DOI: 10.1145/3359789.3359800.

[129] *Usage Statistics of Content Management Systems*. https://w3techs.com/technologies/overview/content_management. [Online; accessed October 2020].

[130] Benjamin VanderSloot et al. "Towards a Complete View of the Certificate Ecosystem". In: *Proceedings of the 2016 Internet Measurement Conference*. IMC '16. Santa Monica, California, USA: ACM, 2016, pp. 543–549. ISBN: 978-1-4503-4526-2. DOI: 10.1145/2987443.2987462.

[131] Kami Vaniea and Yasmeen Rashidi. "Tales of Software Updates: The process of updating software". In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, San Jose, CA, USA, May 7-12, 2016*. 2016, pp. 3215–3226. DOI: 10.1145/2858036.2858303.

[132] Kami E Vaniea, Emilee Rader, and Rick Wash. "Betrayed by updates: how negative experiences affect future security". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM. 2014, pp. 2671–2674.

[133] Francesco Vitale et al. "High Costs and Small Benefits: A Field Study of How Users Experience Operating System Upgrades". In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. CHI '17. Denver, Colorado, USA: ACM, 2017, pp. 4242–4253. ISBN: 978-1-4503-4655-9. DOI: 10.1145/3025453.3025509.

[134] Artem Voronkov et al. "Systematic Literature Review on Usability of Firewall Configuration". In: *ACM Comput. Surv.* 50.6 (Dec. 2017), 87:1–87:35. ISSN: 0360-0300. DOI: 10.1145/3130876.

[135] Rick Wash, Emilee Rader, and Chris Fennell. "Can People Self-Report Security Accurately? Agreement Between Self-Report and Behavioral Measures". In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. CHI '17. Denver, Colorado, USA: Association for Computing Machinery, 2017, 2228–2232. ISBN: 9781450346559. DOI: 10.1145/3025453.3025911.

[136] Rick Wash et al. "Out of the Loop: How Automated Software Updates Cause Unintended Security Consequences". In: *10th Symposium On Usable Privacy and Security (SOUPS 2014)*. Menlo Park, CA: USENIX Association, 2014, pp. 89–104. ISBN: 978-1-931971-13-3.

[137] Rick Wash et al. "Understanding Password Choices: How Frequently Entered Passwords Are Re-used across Websites". In: *Twelfth Symposium on Usable Privacy and Security (SOUPS 2016)*. Denver, CO: USENIX Association, June 2016, pp. 175–188. ISBN: 978-1-931971-31-7.

[138] F.J. Wertz et al. *Five Ways of Doing Qualitative Analysis: Phenomenological Psychology, Grounded Theory, Discourse Analysis, Narrative Research, and Intuitive Inquiry*. Jan. 2011.

[139] *What Hackers Do With Compromised WordPress Sites*. https://www.wordfence.com/blog/2016/04/hackers-compromised-wordpress-sites/. [Online; accessed October 2020].

[140] A. Whitten and J.D. Tygar. "Why Johnny can't encrypt: A usability evaluation of PGP 5.0". In: *Proceedings of the 8th USENIX Security Symposium* 99 (Jan. 1999), pp. 169–184. DOI: 169-184.

[141] *Wordpress Homepage*. https://wordpress.com/. [Online; accessed October 2020].

[142] *Wordpress Plugins*. https://de.wordpress.org/plugins/browse/popular/. [Online; accessed October 2020].

[143] Khaled Yakdan et al. "Helping Johnny to Analyze Malware: A Usability-Optimized Decompiler and Malware Analysis User Study". In: *2016 IEEE Symposium on Security and Privacy (SP)*. May 2016, pp. 158–177. DOI: 10.1109/SP.2016.18.

Appendix A

Updates in Companies

A.1 Questionnaire

Information & Consent

Hello, we're Usable Security researchers from the University of Bonn and our mission is to make your challenges with system updates easier. As a first step, we need to understand your experiences and struggles with software updates in a corporate environment. We conducted interviews with seven colleagues of you and condensed interesting themes. This short questionnaire will take about 10 minutes to answer . We know that your time is precious, which is why every tenth participant gets a 3D-print of a model of her/his choice (max. 3x3x3cm and a reasonable model). If you are interested in this form of compensation just leave us your email address in the commentary field at the end. This email address will be stored separately from your answers and will only be used to communicate about your compensation. Please read all questions and instructions carefully. All of your answers will be checked, and your survey may be rejected in the case of inconsistent answers. Your data will be collected and processed in anonymized form, so that no connection to your person can be made. You can stop participating in this study at any time. If you have any questions please contact us via email.

*1. I have read and understood the information provided above and consent to take part in this study.

- I consent
- I do not consent

Demographics & General

*2. How old are you?
Text-input field
3. What ist your gender?
Text-input field

4. In what country do you work?
Text-input field
*5. For how many years have you worked as a professional system administrator?
Text-input field

Job information

All of the questions on this page refer to a specific company. If you currently work as an administrator, please answer these questions about your current company. Instead, if you do not currently work as an administrator, please answer these questions about the last company at which you worked as an administrator.

6. Is this company an IT company (software/hardware development, hosting, ISP, ...)?

- Yes
- No
- Other (please specify): *Text-input field*

7. Which of the following statements best describes your role in this company?

- My primary responsibility was system administration
- My primary responsibility was not system administration, but I spent at least 20% of my time on system administration
- My primary responsibility was not system administration, but I spent between 1% and 19% of my time on system administration
- I did not perform system administration at that company

8. In a few words, what would you consider as your main task in the company you are working at?
Text-input field
9. What is your main task as a system administrator? If it is the same as in the previous answer, please answer: same.
Text-input field
10. What kind of systems do you administer?

- Clients (e.g. workstations)
- Servers
- Mobile Clients (eg. tablet, smartphone)

- Other (please specify): *Text-input field*

* 11. How big is the company you work at as a system administrator?

- less than 10 employees
- up to 50 employees
- up to 250 employees
- more than 250 employees

12. Do you work in a team?

- Yes, as a team leader
- Yes, as a team member
- No
- Other (please specify): *Text-input field*

*13. What kind of job related education did you receive? (e.g. training, certificate, university)
Text-input field
14. Which of the following statements best describes the security-related training you have received concerning system administration?

- I received security-related training for system administration at that company
- I did not receive security-related training for system administration at that company, but I have received such training at a previous company or school
- I have never received security-related training for system administration

Update Process

Please be reminded that we do not collect or store identifying information. In the following we are interested in your honest opinion.
15. Among all software updates you install for operating systems or any other software running on systems, approximately what percentage do you estimate are **security** updates?
Slider [0-100]
16. Within your job as a system administrator, how much effort does it take you to keep the software on your systems up-to-date?
7-point Likert scale from "1 - Nearly none" to "7 - Nearly all my capacity"
17. What pre-deployment steps do you take before installing an update on a live system?

- We install it on a test system.
- We install it on a small number of production systems before deploying it to all systems or to everyone.
- We install it directly on all production systems.
- Other (please specify): *Text-input field*

18. What is the share of security related updates in relation to all updates (in %)?
Slider [0-100]
19. Which of the following statements best describe the update process in the company?

- There is a written document, that formally describes the steps in the update process.
- There is no written document but an informal guideline that is followed in the update process.
- There is no defined update process.

20. What is the typical time-span between the release of an update to the installation in a normal update process?
Text-input field
*21. Please indicate how often the following situations occur:
Table of the following questions, with a 6-point Likert scale from "1 - Never" to "5 - Always" and the option "Not sure", per question.

- I feel that I am not sufficiently trained as an administrator.
- I think of work- related consequences when doing tasks that have, in case of a failure, an impact on my company (e.g. downtime of a service that everyone uses).
- I feel personally responsible for keeping the software on my systems up-to-date.

22. Please indicate how often the following situations occur:
Table of the following questions, with a 6-point Likert scale from "1 - Never" to "5 - Always" and the option "Not sure", per question.

- Stability considerations hinder the installation of an update.
- Risk considerations hinder the installation of an update.
- Performance considerations hinder the installation of an update.

- Priority/time considerations hinder the installation of an update.
- Software updates are prevented because of other software (e.g. dependencies).

23. Please indicate how often the following situations occur:
Table of the following questions, with a 6-point Likert scale from "1 - Never" to "5 - Always" and the option "Not sure", per question.

- System stability considerations are irrelevant to the installation of an update.
- The risk of breaking dependencies hinder the installation of an update.
- A patch that is known to introduce errors hinder the installation of an update.
- Downtimes caused by the update process hinder the installation of an update.
- Lack of information about the changes an update introduced hinder the installation of an update
- Lack of education and knowledge hinder the installation of an update.

24. Please indicate how much you would agree/disagree with the statements.
Table of the following questions, with a 7-point Likert scale from "1 - Strongly disagree" to "4 - Undecided" to "7 - Strongly agree", per question.

- Deploying security updates in a timely manner is important.
- Post-installation problems in a live system are only a minor concern because they don't happen frequently.
- Users often install software without the knowledge of the administrator.

25. Who makes the decision whether to update or not?

- My team.
- Myself.
- My colleague(s).
- My supervisor.
- None of the above, please specify: *Text-input field*

26. Please indicate how often the following situations occur:
Table of the following questions, with a 6-point Likert scale from "1 - Never" to "5 - Always" and the option "Not sure", per question.

- I feel sufficiently trained as an administrator.
- I can oversee the impact an update would have on our systems.
- I can oversee the impact of a failed update on our system.
- I can oversee the security impact of updates on our systems.

Source and Tools

*27. What sources do you use to get information about current system updates?

- Online publications/news (e.g. cnet.com, Hacker News, heise,...)
- Update management software
- (Software) Publisher newsletters
- External services (e.g. a company that is contracted to inform you)
- Mailing lists
- My users
- Other (please specify): *Text-input field*

*28. What ist your main source to get information about current system updates?

- Online publications/news (e.g. cnet.com, Hacker News, heise, ...)
- Update management software
- (Software) Publisher newsletters
- External services (e.g. a company that is contracted to inform you)
- Mailing lists
- My users
- Other (please specify): *Text-input field*

29. Please explain your previous answer:
Text-input field

Thank you!

30. What do you think are the biggest obstacles in the update process?
Text-input field

31. Thank you for your participation! If you have any further comments for us: Don't hesitate to use the textbox!
Text-input field

32 . If you are interested in the 3D model print just leave your email in this field. We will only use this mail for the communication and will not link it to your answers.
Text-input field

A.2 Interview Guidelines

Questions to explore

1. What does the update process look like?
2. What obstacles are there?
3. Who is involved?
4. What is his/her personal experience and assessment?

Introduction

1. How long has he/she done the job? What is the training? What is he/she doing on a daily basis?
2. What are the systems?
3. Does he/she work in a team?
4. What is the scope of his/her actions?
5. What tools are used?

General update process (or a specific update story)

1. How does he/she come in contact with updates?
2. What is the time frame and the process?
3. What tools are used?
4. Who is involved?
5. Where does the information come from?

(Optional) A second story

1. How does he/she come in contact with updates?
2. What is the time frame and the process?
3. What are the tools?
4. Who is involved?
5. Where does the information come from?

End

1. Do they have a fixed update policy?
2. Are there any feelings connected to new updates or the installation?
3. Is he/she aware of potential impacts of not installed update/failures of the installation? (Are there stories?)
4. Are there wishes concerning the process/tools?
5. Questionnaire

Appendix B

Case-Study Material

B.1 Interview questions

1. Which systems are you in contact with during your work?
2. Are you involved in any update processes?
3. In which way are you involved in update processes?
4. What comes to your mind if you think about the updates you are involved in?
5. Do you use tools to simplify your work during updates?

B.2 Questionnaire

1. Please indicate your field of activity.
   [System administration, Development, Project management]
2. Please indicate all technologies in the list for which you are responsible for updates. For each of them, please additionally indicate your coworkers' role with whom you share the responsibility or who are also involved.
3. Which technologies from the list share dependencies which need to be considered when updating?
4. Which of the circumstances from the previous two questions lead to problems? Why?

List of technologies for questions 2 and 3: WordPress, Typo 3, Imperia, Joomla, Limesurvey, Vue.js, Moment.js, node.js, Express.js, Ionic, Symfony, PHP, Zend, Gentoo, Ubuntu, Windows Server, NGINX, Apache, MySQL, MariaDB, PostgreSQL, HAProxy, Varnish, VMWare vCloud, pfSense, GitLab

Appendix C

Update information

C.1 Survey and Results

1. Welcome and thank you for your participation in our research study!

   The goal of our study is to analyze and understand the impact of update-related information and how it helps you in your decision to deploy the update.

   Therefore, we built this short survey based on previous interviews and findings. Please answer the following questions based on your experience and knowledge. Your data will be collected and processed in anonymized form, in a way that no connection to your person can be made.
   The study should take you around 5-10 minutes to complete and your participation is voluntary. You can withdraw at any point during the study, for any reason, and without any prejudice. If you would like to contact the Principal Investigator in the study to discuss this research, please e-mail martius@uni-bonn.de.
   By clicking the button below, you acknowledge that your participation in the study is voluntary, you are 18 years of age, and that you are aware that you may choose to terminate your participation in the study at any time and for any reason.

   - I consent.
   - I do not consent.

2. How old are you?
   Free response

3. What is your gender?
   Free response

4. For how many years have you been working as a professional system administrator?
   Free response

All of the questions on this page refer to a specific company. If you currently work as an administrator, please answer these questions about your current company. If you do not currently work as an administrator, please answer these questions about the last company at which you worked as an administrator.

5. Is this company an IT company?
   - Yes
   - No
   - Other (please specify)
     Free response
6. In what country is this company?
   Free response
7. Which of the following statements best describes your role in this company?
   - My primary responsibility was system administration
   - My primary responsibility was not system administration, but I spent at least 20% of my time on system administration
   - My primary responsibility was not system administration, but I spent between 1% and 19% of my time on system administration
   - I did not perform system administration at that company
8. In a few words, what would you consider as your main task in the company you are working at?
   Free response
9. What is your main task as a system administrator? If it is the same as in the previous answer, please answer: same
   Free response
10. What kind of systems do you administer?
    - Clients
    - Servers
    - Mobile Clients
    - Internet of Things
    - Other (please specify)
      Free response
11. How big is the company you work at as a system administrator?
    - Less than 10 employees

- 11 - 50 employees
- 51 - 100 employees
- 100 - 500 employees
- 501 - 2000 employees
- More than 2000 employees

12. How many machines/devices do you manage?
Slide bar from 0 to 1000+

13. How many updates do you run on the systems that you administer per week?
Slide bar from 0 to 500+

14. What pre-deployment steps do you take before installing an update on a live system?

    - We install it on a test system.
    - We install it on a small number of production systems before deploying it to all systems or to everyone.
    - We install it directly on all production systems.
    - Other (please specify)
    Free response

15. What kind of job related education did you receive? (e.g. training, certificate, university)
Free response

16. Where do you find out about an available update? (Check all that apply)

    - Online forums
    - Security advisories
    - Blogs
    - News
    - Social media
    - RSS feeds
    - Professional mailing lists
    - Project mailing lists
    - Direct notification from vendor
    - Direct notification from customer
    - Third-Party service

- When the software pops up a notification
- Other (please specify)
  Free response

17. Please indicate the percentage of automatically applied updates in relation to all applied updates:
    Slide bar from 0 to 100

18. How often do you read update-related information (including the installation manual) in order whether or not to update for automatic and manual updates?
    Table of the following questions, with a 7-point Likert scale from '1 - Never' to '5 - Always' and the options 'Does not apply' and 'Prefer not to answer'

    - Automatic update
    - Manual update

19. Please indicate how often the following situations occur:
    Table of the following questions, with a 7-point Likert scale from '1 - Never' to '5 - Always' and the option 'Prefer not to answer'

    - There is a lack of update-related information.
    - Lack of information increase the effort to update.
    - I look for additional information not given by the publisher.

20. Where do you look for additional information (Check all that apply)

    - Online forums
    - Security advisories
    - Blogs
    - News
    - Social media
    - RSS feeds
    - Professional mailing lists
    - Enquiry to the vendor
    - Other (please specify) *Free response*

21. Please rate the subjective time available to you to learn about an update:
    Table of the following statement, with a 6-point Likert scale from '1 - No time' to '5 - No time restrictions' and the option 'Prefer not to answer'

    - Time to learn about an update

The following questions refer to the usefulness of specific update-related information. We want to find out how these factors support you in your decision whether or not to update a machine/device/software.

22. Please rate the usefulness of the following general information-related information:
Table of the following statements, with a 6-point Likert scale from '1 - Not useful at all' to '5 - Extremely useful' and the option 'Prefer not to answer'

    - Release Date
    - Release Number
    - Note Number
    - Note Date
    - Purpose of the update

23. Please rate the usefulness of the following release-notes-related information:
Table of the following statements, with a 6-point Likert scale from '1 - Not useful at all' to '5 - Extremely useful' and the option 'Prefer not to answer'

    - Fixed bugs
    - Still existing bugs
    - Steps to reproduce bugs
    - involved components
    - Changed environment (if necessary)
    - Known issues
    - Closed vulnerabilities
    - Update severity (i.e., critical, moderate..)

An update can have an impact on support-level (i.e., for you) and/or on end-user-level. Please answer the following questions that address these two factors.

24. Please rate the usefulness of the following support-impact-related information
Table of the following statements, with a 6-point Likert scale from '1 - Not useful at all' to '5 - Extremely useful' and the option 'Prefer not to answer'

    - Added feature
    - Removed feature
    - Modified handling of a feature
    - Advertising information (i.e. more colorful)

25. Please rate the usefulness of the following end-user-impact-related information
Table of the following statements, with a 6-point Likert scale from '1 - Not useful at all' to '5 - Extremely useful' and the option 'Prefer not to answer'

    - Added feature
    - Removed feature
    - Modified handling of a feature
    - Advertising information (i.e. more colorful)

26. Please rate the usefulness of the following changelog-related information:
Table of the following statements, with a 6-point Likert scale from '1 - Not useful at all' to '5 - Extremely useful' and the option 'Prefer not to answer'

    - Added files
    - Removed files
    - Changed files

27. Please rate the usefulness of the following installation-manual-related information:
Table of the following statements, with a 6-point Likert scale from '1 - Not useful at all' to '5 - Extremely useful' and the option 'Prefer not to answer'

    - Prerequisites (i.e. reboot necessary)
    - Changed/Added/Removed dependencies
    - Update delivery (zip-file, binary..)
    - Installation manual for the update itself
    - Installation manual for required third-party software

28. Please rate the usefulness of the following other information:
Table of the following statements, with a 6-point Likert scale from '1 - Not useful at all' to '5 - Extremely useful' and the option 'Prefer not to answer'

    - Documentation of added or modified features
    - Disclaimers
    - Support contact information

29. Please rate the usefulness of properties of known issues:
Table of the following statements, with a 6-point Likert scale from '1 - Not useful at all' to '5 - Extremely useful' and the option 'Prefer not to answer'

    - Knowing about possible bugs before they occur
    - Having a workaround for bugs

- Knowing that a bug does not impinge our system

30. What else do you want us to know about update-related information not mentioned in the survey?
Free response

Information	Survey	1	2	3	4	5	*	**Median**
Release Date	1	2	6	10	6	17		4
	2	1	5	2	3	6		4
Release Number	1	4	14	13	7	3		3
	2		2	6	4	5		4
Note Number	1	6	17	12	4	2		2
	2	1	2	8	3	3		3
Note Date	1	6	12	18	3	2		3
	2	1	1	10	3	2		3
Purpose of the Update	1		1	1	5	34		5
	2		3		5	9		5
Fixed Bugs	1		2	2	6	31		5
	2			5	5	7		4
Still existing Bugs	1		4	6	13	18		4
	2		1	5	4	7		4
Steps to Reproduce Bug	1	1	8	14	14	4		3
	2		5	4	5	3		3
Involved Components	1		2	14	16	9		4
	2		1	3	8	5		4
Changed Environment	1		4	10	13	14		4
	2		3	2	5	6	1	4
Known Issues	1		1	1	12	27		5
	2			2	6	9		5
Closed Vulnerabilities	1		2	4	9	26		5
	2		1	4	4	8		4
Risk Qualification	1	1	4	11	8	17		4
	2		2	5	3	7		4
Added feature	1	1	1	9	14	16		4
(Support-Impact)	2		4	5	4	4		
Removed feature	1	1	1	7	12	20		4
(Support-Impact)	2		1	4	4	8		4
Modified handling of a feature	1		2	12	16	11		4
(Support-Impact)	2		1	3	9	4		4
Advertising information	1	14	19	3	3	2		2
(Support-Impact)	2	9	4		2	1	1	1
Added feature	1		3	7	15	16		4
(End-User-Impact)	2		2	5	4	6		4
Removed feature	1	1	5	6	7	22		5
(End-User-Impact)	2			3	8	6		4
Modified handling of a feature	1	2	3	8	11	17		4
(End-User-Impact)	2		1	3	9	4		4
Advertising information	1	14	3	12	6	6		3
(End-User-Impact)	2	4	7	3	2		1	2
Added files	1	3	3	13	9	13		4
	2	1	4	3	5	2	2	3
Removed files	1	3	4	12	11	11		4
	2	1	4	1	6	3	2	4
Changed files	1	3	2	14	10	12		4
	2	1	3	4	5	2	2	3
Prerequisites	1		1	1	8	31		5
	2		2	2	1	12		5
Dependencies	1	1		5	10	24	1	5
	2		1	1	4	10	1	5
Update delivery	1		7	15	13	6		3
	2	1	4	3	5	4		4
Installation manual itself	1	1	5	10	11	14		4
	2	1	1	6	4	5		4
Third party	1	3	7	6	10	15		4
	2		1	7	4	5		4
Documentation of features	1	1	2	6	13	17	2	4
	2			5	7	5		4
Disclaimers	1	13	14	8	1	3	2	2
	2	6	4	5	1	1		2
Support contact information	1	1	8	18	8	4	2	3
	2	1	8		4	4		2

TABLE C.1: Overview of the responses to the information-type on a 5-point scale from "1 - Not useful at all" to "5 - Extremely Useful" (* "Prefer not to answer") separated into the two surveys due to the different wording of the question.

C.2 Additional Affinity Diagrams

Appendix D

Let's Encrypt and Certbot

D.1 Survey after both tasks

These are the questions we asked our participants after they finished each task. On a 7-point Likert scale they should rate the task difficulty as well as the TLS-deployment, the certificate acquisition and the web server configuration.

- Please enter your Study ID:
- Had you heard of Let's Encrypt before the study? (only CA-Certbot task)
- Please describe the purpose of "Let's Encrypt" in your own words: (only CA-Certbot task)
- Overall, the task was ...? (Likert)
- Which aspects were particularly difficult / easy?
- Please tell us your opinion of this task regarding the following aspects: Easy to use, Easy to understand, Time consuming, Transparent, Complicated
- Did you successfully complete the TLS configuration task? (Yes, No, Not sure)
- If you didn't finish the TLS configuration task, which steps are still missing to secure the communication?
- Overall, the process of TLS deployment was... (Likert)
- Overall, the process of acquiring a Certificate from a CA was... (Likert)
- Which aspects were particularly difficult?
- Which aspects were particularly easy?
- Overall, the process of configuring the web server to enable HTTPS was... (Likert)
- Which aspects were particularly difficult?
- Which aspects were particularly easy?

D.2 Final survey

In the final survey we asked the participants about their security background and their experience as an administrator and how many web servers they have administered. In addition we asked them to compare the both tasks with respect to the aspects "Easy to use", "Easy to understand", "Time consuming", "Transparent" and "Complexity"

- Please enter your Study ID:
- I have a good understanding of security concepts. (Likert: strongly disagree to strongly agree)
- How often do you ask for help when faced with security problems? (Likert: never to every time)
- How often are you asked for help when somebody is facing security problems? (Likert: never to every time)
- How often have you added security features to projects you were involved in? (Likert: never to every time)
- Are you currently in charge of a web server? (company, private, non-profit association,no)
- Have you ever installed and configured a web server before?
- Have you ever installed and configured SSL/TLS before?
- Have you ever worked as a system administrator?
  - What web servers have you set up before? (e.g. * Apache, nginx,...)
  - How many web servers have you set up before? (0,1,2-5,6-15,$>$ 15)
- Please compare both tasks regarding the following aspects (Likert from "1 - Task 1 was better" over "4 - they were the same" to "7 - Task 2 was better"): Easy to use, Easy to understand, Time consuming, Transparent, Complicated
- In which tasks did you enabled HSTS (HTTP Strict Transport * Security)? (Only in Task 1, Only in Task 2, In both, In none, Not sure)
- Please explain your answer (Why did you enabled it? Why not? Why don't you know?).
- In which tasks have you enabled HPKP (HTTP Public Key Pinning)? (Only in Task 1, Only in Task 2, In both, In none, Not sure)
- Please explain your answer (Why did you enabled it? Why not? Why don't you know?).
- In which tasks have you enabled OCSP-Stapling? (Only in Task 1, Only in Task 2, In both, In none, Not sure)

- Please explain your answer (Why did you enabled it? Why not? Why don't you know?).
- Did you use Mattermost for asking questions?
- If you used Mattermost to ask questions. What was your experience of the process?
  - Do you think that you would have achieved the same result if you had not been able to chat with the support team via Mattermost? (yes, no)
  - Please explain your answer.
- Thank you for answering the questions! If you have any comments or suggestions, please leave them here:

D.3 Pre-screening questions

This document contains the questions we asked in our pre-screening to recruit the participants. Beside some demographic information we asked them to answer bash- and web server-related questions out of which we calculated a score for each correct answer given.

- Please enter your name:
- Please enter your e-mail address, so we can contact you for our study:
- Please enter your age:
- Please enter your gender:
- Which university are you at?
- In which programme are you currently enrolled? (Bachelor of CS, Master of CS, other)
- Your semester:
- How familiar are you in using the bash-shell? (Likert: "Not familiar at all" to "Very familiar"
- Have you ever configured a web server? (yes,no)
- How many years of experience do you have in programming?
- How many years of experience do you have in system administration?
- Which command is used to find out the currently used IPs? (ifconfig, netstat, ipconfig, iptables,I don't know)
- 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 certificate, X.509)
- Which commands restarts the webserver? (sudo service apache2 restart, sudo /etc/init.d/ apache2 restart, sudo service webserver restart, sudo service IIS restart)
- Where are HTML files served by the Apache-Webserver located after default installation? (/usr/share/nginx/www, /etc/www, /var/www, /home/www)
- Which is the best file permission for your private keys on a Linux system? (0777, 0300, 0644, 0600)
- Please rate the security of the following Hash-functions (Likert: "1 - not secure" to "7 - very secure"): Argon, MD5, BCrypt, SHA-1, RC4
- Please describe the purpose of HSTS:
- Certificate Transparency is ... (providing access to the certificates bytecode, a standard for auditing SSL certificates, checking if a server has enabled HTTPS, a framework that helps maintaining the integrity of the SSL certificate system)

D.4 Abbreviated Mattermost Support Playbook

- **Am I forced to use rsa keys? I could use ecdsa if I'm not bound to make use of [Own-CA-domain], as this site only permits rsa-keys. I would request the certificate directly from LE, if you permit.**
  Please use the rsa keys and the Own-CA in this case.
- **In the survey, under "Study ID" shall I enter my ID, that is printed on the paper (in my case XXX), or my normal student ID?**
  Please enter [Study-Id].
- **I completed the task, the portal is available, should I configure the apache in a special way or is the usage of the default configuration acceptable?**
  Since there are no other websites running I think, if it's accessible for everyone it is fine!
- **Must I request a new certificate?**
  Yes, please do so.
- **I'm having a problem connecting to the server i get: Permission denied (publickey). Is it part of the task to resolve this issue?**
  Please use this command: 'ssh -i " /sshkey.pem"
  ubuntu@DOMAINNAME.com'
- **I'm stuck at hosting the files. I'm trying to create a virtual host to host it**

1. Can you tell me, what have you done until now?
2. Have you created a configuration file for apache2 in /etc/apache2/sites-available?
3. Have you enabled the config file with a2ensite?
4. Have you reloaded apache2?
5. Could you please send me the contents of the .conf file?

- **Is the server running or do we need to set it up?**
  This is installed on the machines you connect to with ssh.
- **Cannot press the 'tilde' symbol on keyboard.**
  Please try ALT-Gr in combination with the "plus"-key
- **Is it okay to use my email address for the use of certbot**
  Please read the instructions again carefully.
- **Now I am having trouble with directory as there is no such directory: home/ubuntu/website**
  Please try adding a slash in front of home: home/ubuntu/website
- **How long should one wait for the result of "openssl dhparam -out dhparam.pem 4096"? With bad luck, this can take hours.**
  Our experience with this command has shown that this command is executed within few minutes (< 5).
- **I am trying to install apache using sudo apt-get install apache2 but it won't work.**
  Please configure the apache2 instance on the server. You don't need to install on your client.
- **Is the IP for apache in browser abc.def.ghi.jkl?**
  The IP for the server is [IP-Adress]
- **I cannot copy from home/ubuntu/website/index.html to /var/www/html**
  Please try putting sudo in front of the command.
- **First I have to configure my server for url http://www.sme-company-7.com then I need to use Certificate of authority or can it be done other way?**
  This is up to you. It should work both ways.
- **I am trying to run ./letsencrypt-auto –apache -d www.sme-company-1.com but it is giving error**
  please try these commands: "export LC_ALL="en_US.UTF-8" and "export LC_CTYPE=
  "en_US.UTF-8" and then run it again.
- **Should I use blinded@blinded.com as account email and should I generate a new key?or is a key existent**
  Please use the pre-entered address and create a new key.

D.5 Study description: Realistic scenario with CA-Certbot

These are the scenario letters we handed out to participants that were in the Framing Role-Play-group and had to obtain a certificate with CA-Certbot. Page 1 contains a scenario description with additional information about the task like the command to connect to the AWS-server they had to configure. On the last page we presented them the four tasks they had to do. For each participant we modified the "URL", as well as the company name for his scenario. We blinded the descriptions for double blind review.

STUDY HANDOUT

TLS Configuration

Please imagine that you are the system administrator at **COMPANYNAME**, a medium-sized enterprise. Your company wants to sell their new product, the "HomeAutomator", and your boss instructed you set up the web server that hosts the product page where people can directly buy it.

Please make sure of the following:

1) Users can access **https://www.URL.com** without any warning messages
2) Make your configuration as secure as possible.

If you need any additional information to complete your task, you can contact your supervisor in your company using a chat-client called **Mattermost**:

Environment:

You will get access to a computer with which you will connect to the server using SSH. Currently you see the Apache2 default message when you open **http://www.URL.com** on the server and an error message when opening **https://www.URL.com** .

To connect to the server, please use the following command:

- **ssh -i "~/sshkey.pem" ubuntu@URL.com**

The HTML documents for the website are already stored **on the server** and can be found at:

- **/home/ubuntu/website**

Company Details:

- **COMPANYNAME, UNIVERSITY CITY**
- Person in Charge: **AUTHOR, admin@URL.com**

YOUR TASKS:

Task 1 / 4

- Please connect to the server and host the HTML files found in **/home/ubuntu/website** on the **server**

Task 2 / 4

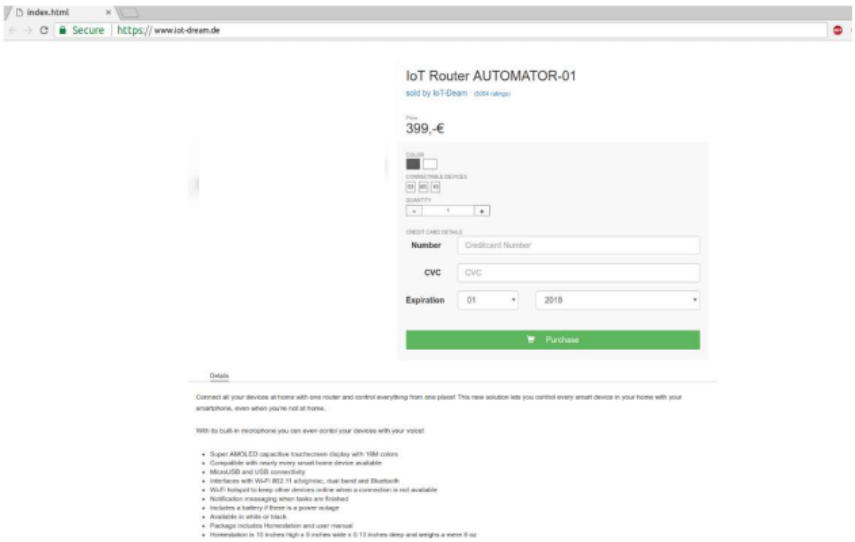
- Configure the web server in a way that you can access the site with the following URLs:
  - **http://www.URL.com**
  - **https://www.URL.com**

- For this, you will have to use a Certificate Authority. Please use https://letsencrypt.org/ to acquire a free CA signed certificate for the server.

- Please use https://www.ssllabs.com/ssltest/index.html to check your configuration

Task 3 / 4

- Please open **https://www.URL.com** and test if it looks like this:



Task 4 / 4

- Please fill out the survey:
  https://www.surveymonkey.de/r/tls-study-le

D.6 Study description: Study scenario with CA-Traditional

This is the scenario for the Framing Study and CA-Traditional group. The structure is very similar to the realistic one except that the task is described without the company scenario.

STUDY HANDOUT

TLS Configuration

You're instructed to set up a web server that hosts a test website.

Please make sure of the following:

1) Users can access **https://www.URL.com** without any warning messages
2) Make your configuration as secure as possible.

If you need any additional information to complete your task, you can contact the study assistant using a chat-client called **Mattermost**:

Environment:

You will get access to a computer with which you will connect to the server using SSH. Currently you see the Apache2 default message when you open **http://www.URL.com** on the server and an error message when opening **https://www.URL.com.**

To connect to the server, please use the following command:

- **ssh -i "~/sshkey.pem" ubuntu@URL.com**

The HTML documents for the website are already stored **on the server** and can be found at:

- **/home/ubuntu/website**

Company Details:

- **UNIVERSITYNAME, UNIVERSITYCITY**
- Person in Charge: **AUTHOR, AUTHOREMAIL**

YOUR TASKS:

Task 1 / 4

- Please connect to the server and host the HTML files found in **/home/ubuntu/website** on the **server**

Task 2 / 4

- Configure the web server in a way that you can access the site with the following URLs:
  - **http://www.URL.com**
  - **https://www.URL.com**

- For this, you will have to use a Certificate Authority. Please use the following CA available at https://www.study-ca.de to acquire a free CA signed certificate for the server.

- Please use https://www.ssllabs.com/ssltest/index.html to check your configuration

Task 3 / 4

- Please open **https://www.URL.com** and test if you get no error message.

Task 4 / 4

- Please fill out the survey:
  https://www.surveymonkey.de/r/tls-study-ca