A Case Study on (Security) Update Processes in Working Environments: Understanding the Context

Maximilian Häring
Fraunhofer FKIE
haering@cs.uni-bonn.de

Christian Tiefenau
University of Bonn
Bonn, Germany
tiefenau@cs.uni-bonn.de

Eva Gerlitz
Fraunhofer FKIE
Bonn, Germany
gerlitz@cs.uni-bonn.de

Ronald Brenner
University of Bonn
Bonn, Germany
s6roremp@uni-bonn.de

Emanuel von Zezschwitz
University of Bonn, Fraunhofer FKIE
Bonn, Germany
zezschwitz@cs.uni-bonn.de

Abstract
Updates are security-relevant and for this reasons should be applied timely, regularly, and routinely to keep systems safe. Yet the process itself can be complex and is often influenced by external factors. Our research aims at investigating the update problem from different perspectives and analyzing the interaction effects of involved stakeholders. In this work, we present the preliminary insights from an ongoing case study focusing on the update process of in-house and customer software in a small (19 employees) tech company. The data was collected in a mixed-methods approach. We combined interviews and surveys (n=8) with the insights from a content analysis of around 300 update related issues from a text-based ticket system. We found that priorities still conflict with the goal of security but also that updates are already rolled out on a regular basis and are part of the organizational work flow. Finally, we reflect on our methodical approach that involved ethnographic aspects.

Author Keywords
updates; it professionals; developers; security; security information workers

ACM Classification Keywords
H.5.m [Information interfaces and presentation (e.g., HCI)]: Miscellaneous

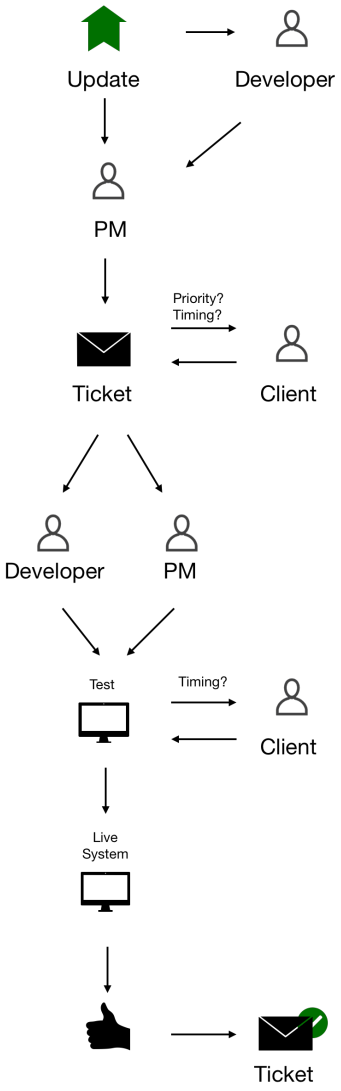

Figure 1: Sketch of the update process.

Introduction

Software updates affect everyone working with digital computer systems. Among other reasons (e.g., feature upgrades) they are used to provide security patches to software. Anecdotal evidence from the past shows that shortfalls can have a big impact [1] and put business data and client data at risk. Previous work has already provided insights about the security-critical aspects of human update behaviour [5, 3].

E.g., Vaniea at al. [5] studied end users through surveys asking for contrasting experiences from the participants. They found that common update processes can be structured in six phases. In addition, they presented common reasons for users to not update, such as satisfaction with the current version or general mistrust towards updates. Mathur et al. [3] also used a survey as an instrument to gather data but focused more on quantifying the prevalence of beliefs. The authors name three categories of beliefs (necessity, cost and risk) that often hinder users to update.

In the area of administrators and developers, as a specific population of professional users, the literature covering updates is sparse. We want to fill this gap and present a case study that investigates the interaction effects of different stakeholders in the update process.

We “joined” a small, 19-employee company, from now on called DevComp, and followed an ethnographic approach to understand the different perspectives on the update problem. One of the authors was working for DevComp as a developer and performed the interviews, surveys and analysis of the ticket system. DevComp usually takes the role of a service provider, meaning that DevComp develops web sites and applications on the basis of common frameworks, such as Wordpress, and then supports their operation. The service includes the integration of new features as well as commissioned bug fixing and the deployment of updates. At the time of the study, DevComp was involved in more than 100 ongoing projects of different sizes. While previous work relied mainly on self-reported data, we are - to the best of our knowledge - the first to perform a contextual inquiry in the context of software updates. In addition, we are not observing single system administrators, developers or end users but all relevant stakeholders from the respective company. Because of the complexity in this field, there is a large number of potential parameters that influence the measurement and the analysis process. So, identifying interesting patterns is a challenge and part of this project. In this work we present and evaluate the preliminary findings and discuss our research approach. Finally, we give an outlook on the next steps, where we want to gather contextual in-situ information to enrich our quantitative data in a qualitative way to get additional insights into the security behavior of this population.

Our focus for this project are the following **Research questions**

1. Which relevant stakeholders do exist?
2. How do the stakeholders see the process from their perspective?
3. How does the customer influence the update process?
4. How do the stakeholders deal with problems during the update process?
5. Finally: How can we as usable security experts gain insights from this data? How can processes be improved?

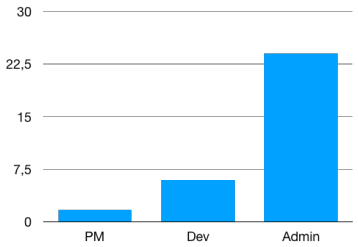

Figure 2: Average of the reported dependencies that have to be thought of while updating.

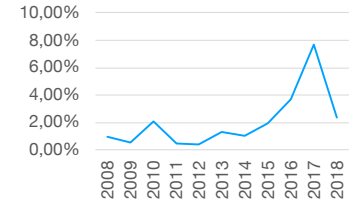

Figure 3: Percentage of update related tickets in the programming category.

Pre-study

We had a two step approach: 1) Interviews (n=19) with the employees and surveys (n=8) with the identified stakeholders, 2) content analysis of a text-based issue tracker. In the following section we present our research approach in more detail and the preliminary results.

Interviews and Surveys

Approach We started the process by conducting unstructured interviews (about 5-10 minutes) with all employees. This was done to identify those who are connected in any sort of way to software update processes. In addition, it gave us a first picture of the overall standard process for updating in-house and client software. Following the interviews, we surveyed the identified stakeholders in DevComp to identify their technical responsibilities.

Results Eight of the 19 employees were identified to be part of the different update processes. These eight DevComp employees cover various roles from software developers (n=4) and project managers (n=3) to one administrator who monitors and manages the servers. Figure 1 shows the standard process that is usually followed by the company. During an update process, normally more than one stakeholder is involved. We want to direct the attention to the situation that updating is not only carried out by developers, but can also be done by project managers that have fewer technical knowledge and probably different mental models of the underlying technologies. We found that none of the project managers had an IT background. Their IT knowledge is based on past experiences at the work place. In our survey, we asked about the number of technical dependencies, they have to think about while updating. The answers were diverse as can be seen in Figure 2.

We asked who initiates the updates and who pays for it and found that this service is usually part of the monthly billed service contract. This contract gives DevComp the possibility to invest time into the updating of the projects, but is also used to give support for the project. While budget could be a restriction, participants reported that their time is the restricting factor. Other things are prioritized higher like a new search function with a deadline. If there is no contract or the budgeted time is exhausted, each case has to be handled separately and time and costs have to be calculated and discussed with the customer. Information for updates comes through software itself, mailing lists or media to the employees.

Ticket System Analysis

Approach From our initial interviews, we learned that DevComp used a ticket system for project management internally and also externally, e.g., for communication with customers. We gained access to all tickets from a period of 10 years (2008-2018). The data base contained approx. 30,000 tickets. Around 12,500 were in a category representing programming tasks, which also involved all update-related issues. We filtered the tickets based on the terms "update", "upgrade" and "patch" and then manually went through the list to filter out tickets that were not connected to an update process, e.g., for subjects as "update to the current status".

Results We ended up with 295 update related tickets. A ticket contains fields like the project name, person in charge, the category, a textual summary, a priority field and some more details like deadlines or effort estimations. We found out that most of the time the status field was not actively used. It provides only basic information about the current state of the ticket, i.e., if it is closed or open. In addition, we found that the priority field was not correlated to the importance of an issue in terms of security, but it usually

	Programming	Update
Min	0	0
Max	2100	723
Avg.	62	67

Table 1: Time in days tickets are processed.

Figure 4: Number of different authors that left a note on an update ticket.

indicated how fast it should be handled based on other external factors (e.g., pressure through the customer or deadlines for launches). We found that the circumstance around the ticket system changed over the course of the years. E.g., the average time to process a programming ticket was reduced from 195 days in 2008 to 45 days in 2018 which might be due to the introduction of support contracts, as this leads to fewer administrative work. It is also likely that other factors influence these numbers. The average time to process an update related ticket in 2018 was 54 days and 338 days in 2008, though there was only one such ticket in this year. The amount of update related tickets vary per year. From a minimum of one in 2008 to a maximum of 110 in 2017.

In Figure 3 we depicted the percentage of update related tickets, in relation to the amount of programming tickets. This includes one event (in 2017) we want to highlight, as it strongly increased the number of update tickets that year. It was an attempt of making a big project based on PHP 5.3 compatible to a newer version (7.0), as the developer of PHP had stopped their support in terms of security-updates for the old version [2]. Eventually, the procedure was set on hold. We could not identify specific reasons for this step.

A summary of how long the tickets were open can be seen in Table 1. Some tickets represent reoccurring tasks and will never get closed.

We could confirm our finding from the interviews that updating is a process which involves more than one stakeholder. It is not uncommon (47%) that more than two different persons leave a note in an update issue. A more detailed picture of this can be seen in Figure 4.

We want to point out that we also observed update cases that are not covered by the ticket system process. Such as the administrator was not using the ticket system for himself, but his own internal management for weekly tasks. Security updates for servers, internal and those for customer projects, (i.e., web server, data base) are installed automatically. In addition to the documented cases there are many small patch-level updates that go on undocumented done by developers. Every project has an assigned team which is responsible to keep the software packages, i.e. frameworks such as Symfony [4], up to date in a weekly routine.

Discussion and Future Work

We got insights into the update processes of the participating company. We interviewed employees and explored the ticket system as a way to gather update-related data post-hoc. Many questions we had at a specific point could not be answered by that approach alone. We found that time consumption and priorities hinder the deployment of updates, but also that many updates are already, if possible, rolled out on a regular basis. With our on-sight researcher we were able to gather contextual and historical insights, e.g. through short spontaneous in-situ interviews.

As a follow up study we want to set the focus on how and why things happen like they do, observing the participants. Where do project managers fail while tackling the update task by themselves and where do developers struggle and spend most time on?

At last we also want to know if the ticket analysis approach can be used for other companies with different settings where we do not have the possibility of installing on-site researchers. Therefore we are currently deploying a study framework which combines content-analysis of the ticket system with experience sampling. We aim at getting information about the specific problems the person who updates runs into.

REFERENCES

1. Electronic Privacy Information Center. 2019. Equifax Data Breach. https://epic.org/privacy/data-breach/equifax/. (2019). Accessed: 2019-05-27.
2. The PHP Group. 2019. Unsupported Branches. https://www.php.net/eol.php. (2019). Accessed: 2019-05-31.
3. Arunesh Mathur, Nathan Malkin, Marian Harbach, Eyal Péer, and Serge Egelman. 2018. Quantifying Users' Beliefs about Software Updates. *CoRR* abs/1805.04594 (2018). http://arxiv.org/abs/1805.04594
4. Symfony SAS. 2019. Symfony. https://symfony.com/. (2019). Accessed: 2019-05-31.
5. Kami Vaniea and Yasmeen Rashidi. 2016. Tales of Software Updates: The Process of Updating Software. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 3215–3226. DOI: http://dx.doi.org/10.1145/2858036.2858303