Human Factors in Cybersecurity, Vol. 53, 2022, 111–119
https://doi.org/10.54941/ahfe1002208

A Software Security Study Among German Developers, Product Owners, and Managers

Stefan Dziwok, Sven Merschjohann, and Thorsten Koch

Fraunhofer IEM, Zukunftsmeile 1, 33102 Paderborn, Germany

ABSTRACT

Online news portals report almost daily on security incidents in all kinds of software products in finance, health, and engineering. Moreover, multiple security reports conclude that there is a growing number of security vulnerabilities, attacks, and incidents. This raises the question of the extent to which companies address software security while developing and operating their products. This paper reports on the results of an extensive study among developers, product owners, and managers in Germany. Our results show that ensuring security is a multi-faceted challenge for German companies, involving low awareness, inaccurate self-assessment, and a lack of competence on the topic of secure software development among all stakeholders. Thus, there is an urgent need to improve the current situation.

Keywords: Software security, Study, Developer, Product owner, Manager, Awareness, Security tools, Training

INTRODUCTION

The security requirements for software-intensive systems are constantly rising as software-intensive systems and apps increasingly process critical data in the domains of finance, health, and engineering. Nevertheless, online news portals report almost daily on security incidents in all kinds of software products all over the world (threatpost 2021; BBC 2021; Cyware 2021) and multiple security reports conclude that there is a growing number of security vulnerabilities, attacks, and incidents (Verizon 2021; Morgan 2017). An impressive example is given by (NIST 2021): In 2020, they collected on average 50 new vulnerabilities in software products daily, which results in 16,132 vulnerabilities for the whole year. This has led us to the research question to which extent companies address software security when developing and operating their software products. Since we, the authors of this paper, are from Germany, we have initially conducted an extensive study that focuses on German companies. In contrast to related works, we did not only survey the security awareness and competence of the software developers of various companies but also of their product owners and managers, as we wanted to obtain a holistic picture of the current situation across all involved roles.

The result of our study is that ensuring secure software products is a multi-layered challenge for companies, and there is a need for action. Developers,

product owners, and managers alike lack awareness regarding security and miss important security competence for their specific roles. In addition, the developers report that they are not satisfied with their existent processes and tools and that they often do not use security tools at all. Concerning product owners and managers, we identified that nearly all of them state that security is important but not important enough to take proactive measures. For example, product owners place too few or no security requirements on the product in development. Additionally, managers only rarely implement measures to increase the security competence of their staff. Thus, we conclude that many companies risk that their products are not sufficiently protected against malicious attacks. Fortunately, many developers, managers, and product owners know their current challenges and are willing to improve the current situation.

While our complete study is available as a whitepaper (Dziwok et al. 2021), this paper shall concisely summarize the study: First, we discuss related work. Then, we explain our methodology and state the highlights of our study. Afterward, we list our threats to validity. Finally, we conclude our paper and give an outlook to future work.

Related Work

This section provides a brief overview of relevant empirical studies on the use of secure software development practices among developers, product owners, and managers.

Many studies focus on security tools and their usage during the software development lifecycle. These typically evaluate specific tools to identify areas of necessary improvements, e.g., (Vassallo et al. 2018; Christakis and Bird 2016; Nguyen Quang Do et al. 2020). However, these tool studies mostly focus on the tools and their usage by the developers, but not on the whole secure software development processes like we do.

In the broad field of security, there are several studies available having a look at the situation from the consumer or company perspective and how they perceive the greatest threats and challenges concerning the security, e.g., the German Bitkom's yearly Trust and IT-Security study (Bitkom 2021) or World Economic Forum's yearly global risk report (World Economic Forum 2021). However, we found that only a few studies exist that focus on application security during the software development process of companies. For instance, (Karim et al. 2016) did a case study in Saudi Arabia. They interviewed 4 participants and surveyed another 15 participants in various roles about their current security activities already in place to assess their model for enterprise security adoption. Another example is (Rindell et al. 2021), who conducted an online survey among 62 software security experts in Finland and concluded that regulations are one of the driving forces behind security engineering. In addition, (Rindell et al. 2021) found that as the adaptation of agile methods increases, the adaptation of security activities also increases.

Lastly, while not focusing on the actual survey, the BSIMM as a software security maturity model for companies, also surveys the current state of different companies and aggregates this to a scale of zero to five in several different categories.

In contrast to the existing work, we were interested in, how the view, competence, and awareness on software security differ between software developers, product owners and managers, which no current studies examined yet.

METHODOLOGY

In this section, we present the methodology of our study consisting of an online survey and semi-structured interviews.

Online Survey Among Software Engineers

Population: We invited participants from all roles involved in the software development including developers, product owners, and managers. We used three ways to gather participants: First, we used our direct contacts from the industry and asked them to invite their teams internally. Second, we created posts on our institution's social media channels and website. Third, the survey was promoted by the media of the publishing house Heise and among several company networks. In total, we received responses from 365 participants. We excluded all responses that were incomplete, answered in an unrealistically short time, or not from Germany. After this filtering, we gathered 256 responses.

Data collection: We conducted the survey using the online tool Survey Monkey. The survey was open for six weeks. On average, the participants needed 25 minutes to complete the questionnaire, measured based on the session duration per participant collected by Survey Monkey.

Design: Initially, we conducted a literature search to identify relevant related work. Unfortunately, none of the existing studies provided a survey instrument (i.e., questionnaire) that can be reused. Hence, we created a new questionnaire for a cross-sectional survey. Five researchers created and selected the questions in a top-down process, starting from the research questions and breaking them into concrete ones. The questionnaire was reviewed by three more researchers and then modified based on the feedback. Finally, we conducted a test phase: First, we performed two internal tests with students from our research group to verify the clarity of the questions and measure the time needed to complete the questionnaire. Second, we performed three external tests with industry professionals involved in software development.

Interviews Among Product Owners and Managers

Population: We performed 17 interviews with product owners and managers (who have personnel responsibility) from German companies. Four of them were our previously known contacts. In addition, we invited several randomly chosen companies from our region and used the first-come, first-served principle to conduct the interviews with persons that volunteered to participate. As a result, seven interviewees were product owners, six were managers, and four had both roles. All experts are involved in software development during their professional work.

Data collection: Two researchers performed each interview. One researcher was the moderator asking the questions, and the other one wrote a

protocol and, in rare cases, asked questions. Additionally, an audio recording of all sessions was made. After the interviews, the recordings were automatically transcribed and used to extend the protocols created during the interview. On average, each interview took 45 minutes.

Design: We applied a similar process as the survey to design the questionnaire used as a guide during the interviews. We created one version for each role, which differ only in a few questions. The experts who had both parts were asked all questions.

KEY FINDINGS OF THE STUDY

In this section, we first report about the background of our participants. We continue with explaining our findings concerning processes and tooling. Then, we report about the security competence of all participants and their current actions to expand the competence. Finally, we estimate the awareness of the participants.

Professional Background of the Participants

Of the 256 developers, 61% state that they have more than ten years of professional experience in software development. 18% of the developers have between six and ten years, 15% have between two and five years, and 5% have less than two years of experience. Thus, we mainly reached experienced developers with our online survey. Overall, 40% of the developers work at small and medium-sized enterprises. The companies' business models are very different: software is used within the company (55%), is licensed to customers (36%), and/or developed directly for customers (28%). 15% of the developers are also sent to other companies. The developers work in various domains: web (66%) and backend (59%) are the main focus, followed by desktop (37%), mobile (25%), and embedded (14%). The most popular IDEs are IntelliJ, Eclipse, Visual Studio, and VS Code. Java, JavaScript/TypeScript, and C# are the most used programming languages. On average, the developers use two different development environments and two to three programming languages.

The interviewed product owners and managers are from small (2 people), medium-sized (10 people), and large (> 250 employees) companies (5 people). They develop software for various industry sectors (e.g., automotive, healthcare, and insurance). Additionally, almost half of them state that they develop software for more than one industry sector. Their companies' business models also differ a lot: software is developed for internal use, direct customer order, or licensed sale. Two interviewees also work for companies that send their employees to other companies.

Processes

The developers agree that the current processes for secure software development and operation need improvement: In each discipline (requirements, design, implementation & test, and release), most developers (~80%) desire more understandable and precise processes. Moreover, 64% of the developers

think that not enough time is invested in their team for secure software development. In addition, most of them use only very few measures (templates, standards, experts, reviews, etc.) for secure software development. Furthermore, it is also worrying that 20% of the developers admit to not paying attention to security during implementation and testing at all. Nevertheless, 62% of the developers believe that their overall development process and the associated tools are suitable for their needs. We will address this fact further when discussing about the awareness.

Our interviews with product owners and managers reveal that security has a rather low to medium priority in their software development processes. The majority refers to the pen test, which only occurs after development. However, explicitly considering security during development is very rare. In addition, most product owners state that security plays only a minor or no role at all in their agile meetings (planning, retro, review). Thus, security is typically handled unsystematically in the software development process (i.e., without the establishment of concrete measures), which results in an increased risk for the quality of the resulting products.

Tools

Tools are an essential component of a successful, secure software development. The developers, product owners, and managers also see the importance of this topic. However, our study has shown that the distribution and use of tools are shallow, especially in the early development disciplines. Even during implementation and operation, a large proportion does not use tools for secure software development. As a result, many errors are discovered late in development or after release, necessitating expensive and time-consuming repairs. In addition, the risk of security incidents occurring in productive use increases.

Another interesting result of the study is that the developers have a high unmet need for tools, e.g., 72% think that more or better tools would help them to perform their tasks better during implementation. The interviewed product owners and managers agree that there is a high demand for suitable tools for secure software development among their developers and are open to purchasing and introducing them. Noteworthy, according to the product owners and managers, they typically have a sufficient budget for tools, but their developers did not approach them until now for purchasing new tools.

Security Competence

Our study found that the developers' competencies are often too low and very diverse. For example, the developers' self-assessment shows that most of them are unfamiliar with the given security topics and have very little practical knowledge. Two-thirds of the developers also think that the current skills of their team are not sufficient to develop or operate software securely. However, more than two-thirds of the developers believe that all team members should have a high level of competence.

Many product owners and managers do not have sufficient competencies to fulfill their tasks. Most product owners and managers also see it this way, as two-thirds would like to have more competencies themselves.

Concerning their developers, the product owners and managers have only few requirements for them, and these requirements differ a lot. Most product owners and managers can only state implementation-specific requirements. The other disciplines seem to be significantly less in focus for them. Nevertheless, more than half of the product owners and managers think that their developers need to expand their competencies in secure software development.

Competence Expansion

Most developers, product owners, and managers are not aware of trainings offered in German-speaking countries even though most stated before that expanding security competencies is necessary. In addition, two-thirds of the developers who are aware of the trainings on offer are not satisfied with them.

Another survey result is that only 55% of the developers regularly inform themselves about (new) potential security vulnerabilities. Hereby, online news sites like Heise are their most mentioned sources. Moreover, only 23% of the developers regularly attend meetups and conferences on security topics. In our interviews, the managers and product owners confirm that their developers only rarely attend such venues.

Furthermore, it has become clear in our study that the developers want different formats to expand the competences are necessary, e.g., self-studies, on-site training, and remote trainings.

Another finding is that the product owners and managers actively encourage their developers to increase their general software development competence through training courses, conferences, or self-study. However, it became clear that the product owners and managers do not systematically promote or demand software security trainings among their developers. Furthermore, most of them do not know whether their developers participate in security trainings. A rarely mentioned exception to that is the measure to appoint and train security champions (Tondel et al. 2020). These are developers that have high security competence that remain in their team such that the team itself can assure the security of its product. Thereby, these champions act as multipliers because they shall not be the single person focusing on security – instead, they shall encourage the team that everyone shall take security into account.

Awareness

Most developers are only on a low awareness level – the interviewed managers and product owners confirm this. However, a minority of the developers is either not at all or comprehensively aware. In addition, we identified an inaccurate self-assessment of many developers: For each discipline, the developers state that they pay attention to security, but they also state that measures (templates, standards, procedures, tooling, experts, reviews) rarely

exist. Without these measures a systematic secure software development is very unlikely.

Most product owners and managers have a low awareness level as well. The predominant attitude is that security is necessary but not important enough to be addressed systematically and with high priority in processes, tools, and competence development. However, this would be in our opinion a necessary task due to the results of our study. Additionally, all product owners and managers are aware of the need of data protection, primarily due to the GDPR, but rarely aware of the security or the risk of internal perpetrators when their products are used non-publicly. Moreover, security-aware and committed product owners and managers exist as well but are frequently confronted with a lack of understanding of colleagues and customers, resistance from superiors, or rigid processes.

THREATS TO VALIDITY

In the following, we discuss the construct validity, the external validity, and the reliability based on the guidelines of (Runeson et al. 2012).

Construct Validity: The questions used in the survey and the interviews are the outcome from several workshops where software security researchers and practitioners with medium to high security expertise participated. We avoided the possibility that the interviewers ask wrong or irritating questions as we – the researchers that conducted this study – held the interviews by ourselves. For gaining a representative survey, we made sure that developers were selected as random as possible. Most developers of our survey were invited by a news article from the German publishing house Heise and only a minority were developers of our industry partners. Our industry partners have small to big companies in all branches of the IT. All interview partners came from partners of Fraunhofer IEM. Thus, they were not chosen fully randomly. However, these companies differ significantly in their size, their domain, and their business model. Moreover, we knew only 3 of the 17 interviewed persons upfront. Thus, the majority was not influenced upfront by us or our project AppSecure.nrw. We pretested our survey and our interviews with people from our target group to identify whether our questions were understandable and are interpreted as intended by us and made changes due to this pre-test.

External Validity: As our study was conducted with companies from Germany only, it might be the case that our results are not applicable to companies outside of Germany as they have other standards and laws that they must consider. Though, several international standards like ISO IEC 27001 exist. Even though we made a pre-test to analyze whether our questions are interpreted correctly, it still might be the case that the participants of our survey misinterpreted our questions. In our semi-structured interviews, this possibility is even smaller as our interviewed people always had the chance to ask questions by themselves. Our developer survey is representative for our intended target group as 256 people have answered it and 900.000 developers exist in Germany (z-score: 1,96). However, our number of interviews

among product owners and managers might not be sufficient to extract a representative result as we only interviewed 17 persons.

Reliability: The questions of our study (survey and interviews) are based on our experience concerning secure software engineering. Other researchers might ask these questions differently. This is especially the case for our semi-structured interviews as the follow-up questions depend on the interviewer. Moreover, we might have made mistakes while analyzing the interviews, especially when matching the interview answers to our pre-defined classes and when interpreting the complete encoding of our interviews. Also, we might have made mistakes while analyzing the survey, especially when analyzing the free text fields or while drawing conclusions. To prevent all these mistakes, we peer-reviewed all our work. To minimize human errors while analyzing our survey, we used tools where possible, e.g., we used Survey Monkey for automatically collecting and exporting the survey data, we recorded all interviews, and we used a reliable voice-to-text software (AmberScript) for the transcription. Finally, using a self-created script, we automatically processed all raw data wherever possible.

CONCLUSION AND FUTURE WORK

Our study has shown that ensuring security is a multi-layered challenge for companies in Germany. We identified the following three key findings. Firstly, most developers, product owners, and managers, have only a low level of awareness: they feel security is essential, but they do not act accordingly. Many study participants also lack awareness regarding security and internal perpetrators. Secondly, most developers have an inaccurate self-assessment on security: They think that they pay attention to the subject of security - but state that they do not have the appropriate measures, processes, internal experts, and tools. And thirdly, all participants in software development need more competence (knowledge and skills) in the subject of secure software development. Although almost all participants in the study would like to see an increase in competence, this has so far only happened sporadically and unsystematically and with little to no support by product owners or managers. Fortunately, nearly all study participants want to increase their competency.

To conclude, the current situation harms the security of software products in the medium and long term. The combination of low awareness and inaccurate self-assessment of all those involved in software development means that the current state is perceived as sufficient. Thus, no improvement (e.g., a systematization of processes and expansion of competencies) is currently sought. However, the steady increase in attacks and the numerous security incidents in recent years show that ensuring security is becoming increasingly essential to avert dangers for companies and customers.

We plan to work on various topics: We will explore how companies and their development teams can introduce security-enhancing activities in their processes so that all involved persons universally accept them – one idea that we are currently investigating is a new software security maturity model for

agile teams. Moreover, we want to make the value for security more transparent such that decision-makers like managers and product owners understand the importance of security-related activities. In addition, we are currently developing role-specific software security training for product owners and managers, as these do not exist yet. Finally, we want to execute studies outside of Germany to examine the current state in different nations with their specific culture, laws, education, and industry branches.

ACKNOWLEDGMENT

This work has been developed in the research project "AppSecure.nrw - Security-by-Design of Java-based Applications" funded by the European Regional Development Fund (ERDF-0801379).

REFERENCES

Bitkom Research (2021): Vertrauen und IT-Sicherheit 2021. Bitkom e.V., online https://www.bitkom-research.de/en/product/143.

BBC. (2021): News: Cyber-Security. https://www.bbc.com/news/topics/cz4pr2gd85qt/cyber-security.

Christakis, M., Bird, C. (2016). What Developers Want and Need from Program Analysis: An Empirical Study. Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering (ASE '16), pp. 332–343. ACM, NY, USA.

Cyware (2021): Latest Cybersecurity News and Articles. https://cyware.com/cyber-security-news-articles

Dziwok, S., Koch, T., Merschjohann, S., Budweg, B., Leuer, S. (2021). AppSecure.nrw software security study. http://arxiv.org/pdf/2108.11752v1

Karim, N. S. A., Albuolayan, A., Saba, T. & Rehman, A. (2016). The practice of secure software development in SDLC: an investigation through existing model and a case study. Security and Communication Networks

McGraw, G. and Chess, B. (2009). The building security in maturity model (bsimm).

Morgan, S. (2017). Top 2016 cybersecurity reports out from at&t, cisco, dell, google, ibm, mcafee, symantec and verizon.

Nguyen Quang Do, L., Wright, J., and Ali, K. (2020). Why Do Software Developers Use Static Analysis Tools? A User-Centered Study of Developer Needs and Motivations. IEEE Transactions on Software Engineering.

NIST (2021). Nvd: National vulnerability database: Vulnerabilities per year.

Rindell, K., Ruohonen, J., Holvitie, J., Hyrynsalmi, S., and Leppänen, V. (2021). Security in agile software development: A practitioner survey. Information and Software Technology, Volume 131

Runeson, P., Host, M., Rainer, A., and Regnell, B. (2012). Case Study Research in Software Engineering: Guidelines and examples.

Tondel, I. A., Gilje Jaatun, M., Soares Cruzes, D. (2020). IT Security Is From Mars, Software Security Is From Venus. IEEE Security & Privacy, vol. 18, no. 4, pp. 48–54, doi: 10.1109/MSEC.2020.2969064

threatpost (2021). News. https://threatpost.com

Vassallo, C., Panichella, S., Palomba, F., Proksch, S., Zaidman, A., and Gall, H. C. (2018). Context is king: The developer perspective on the usage of static analysis tools. IEEE 25th International Conference on Software Analysis, Evolution and Reengineering (SANER) pp. 38–49.

Verizon (2021). DBIR: 2021 Data Breach Investigations Report. https://www.verizon.com/business/resources/reports/dbir/

World Economic Forum (2021). The Global Risk Report 2021: Insight report. World Economic.