Lessons Learned from Recruiting Participants with Programming Skills for Empirical Privacy and Security Studies

Lessons Learned From Recruiting Participants With Programming Skills for Empirical Privacy and Security Studies

Mohammad Tahaei
mohammad.tahaei@bristol.ac.uk
Department of Computer Science
University of Bristol
United Kingdom

Kami Vaniea
kami.vaniea@ed.ac.uk
School of Informatics
University of Edinburgh
United Kingdom

ABSTRACT

In the past decade the usable privacy and security research community has extended its research to expert users such as software developers who have been shown to face challenges when working with privacy and security technologies. However, it is often challenging to run empirical studies with developers because they can be hard to recruit for interviews or surveys. Researchers have successfully used social media, crowdsourcing platforms, and computer science students for recruiting, but recruitment is still a significant pain point in many studies. This short paper reflects on our experience recruiting developers, particularly for empirical privacy and security research, for multiple studies from 2019 to 2022.

CCS CONCEPTS

• **Security and privacy → Software and application security**; **Human and societal aspects of security and privacy**; **Usability in security and privacy**; • **Human-centered computing → Human computer interaction (HCI)**; **HCI design and evaluation methods**; • **Software and its engineering**;

KEYWORDS

recruitment, developers, crowdsourcing, usable privacy and security, empirical software engineering

ACM Reference Format:
Mohammad Tahaei and Kami Vaniea. 2022. Lessons Learned From Recruiting Participants With Programming Skills for Empirical Privacy and Security Studies. In *Proceedings of 1st International Workshop on Recruiting Participants for Empirical Software Engineering, co-located with the 44th International Conference on Software Engineering (RoPES - ICSE 2022).* ACM, New York, NY, USA, 3 pages.

1 AUTHORS' EXPERIENCES

Starting in 2019, we have recruited participants with a background in privacy, security, and programming for multiple studies (Table 1). The experience level of participants has ranged from novice to experienced participants in large tech companies. Methodologies used are also varied and include interviews, tasked-based surveys, and artifact analysis (e.g., developers' posts on Stack Overflow).

2 LESSONS LEARNED FROM RECRUITMENT

Trade-Offs Between Researchers' Time, Participants' Prior Experience, and Number of Participants. We find that social media such as Twitter and LinkedIn may help recruit experienced participants. However, it comes with a trade-off that it requires several months and extra effort to search for potential candidates, answer their questions, and results in only a few participants. In one study, we spent several months recruiting on multiple social media channels, which resulted in 12 experienced participants [2].

Virtual vs. In-Person Studies. While virtual studies allowed us to recruit participants worldwide, they also created a barrier for running lab studies and observing participants. Virtual meetings are easy to cancel too. For example, several participants canceled their appointments [2], but we did not have such experiences with in-person meetings [3], encouraging them to commit to showing up. Therefore, we suggest virtual studies consider recruiting more participants in the case of a high no-show rate.

Computer Science Students as a Go-To Solution. Computer science students were easy to recruit. In an interview study intended to understand students' security mindsets, we recruited 20 students in less than a month. Scheduling the interviews was also easier because everyone was located on the same campus, making it easy to find and attend the session [3].

Consider Using Crowdsourcing Platforms. Crowdsourcing platforms can provide a large pool of potential participants. In a study designed to understand opportunities for recruiting participants with programming skills [5], we found that Appen is not designed for recruiting this type of participant; Clickworker may be helpful, but its interfaces are not designed for surveys and recruitment; and MTurk provided few workers with actual programming skills. Prolific is the only platform that might result in a decent number of participants who pass the programming questions. We also find that computer science students are a reasonable group for finding participants with programming skills.

Make Use of Publicly Accessible Forums' Data. A quasi-recruitment method for understanding developers is to analyze their artifacts. We have done two studies with Stack Overflow's data, and this method allows for the study of a whole community and gives insights into developers' practices on a large scale. We encourage more studies with developer forums, potentially across multiple forums, and combine those studies with interviews and surveys to produce a strong theory and study design plan before recruiting developers, which is more time-consuming.

Check Terms and Policies of Platforms. While we have recruited participants by harvesting emails from GitHub, we believe that the method is not sustainable long term. It is not encouraged by the platform's terms of service and privacy policy to harvest emails for

RoPES - ICSE 2022, May 21–29, 2022, Pittsburgh, PA, USA

Table 1: A summary of our studies from 2019 to 2022.

Publication	Type of participants	Recruitment channel	To study . . .	Method	N	Year
[3]	Computer science students	Local University mailing list	Security mindsets of participants	Interview, qual.	20	2019
[7]	A range of developers	Stack Overflow	Developers privacy challenges and how they conceptualize privacy	Artifact, qual. & quant.	315	2020
[6]	A range of developers	Prolific and GitHub emails	Usefulness of security notifications in static analysis tools	Survey, qual. & quant.	132	2020
[2]	Experts in privacy	Social media, word of mouth	How experts champion for privacy in software teams	Interview, qual.	12	2020
[1]	A range of developers	Prolific, GitHub emails, and word of mouth	Impact of interfaces and choices on developers' privacy decisions	Survey, qual. & quant.	400	2021
[4]	A range of developers	Stack Overflow	Developers' privacy-related advice	Artifact, qual. & quant.	119	2022
[5]	Participants who claimed to have programming skills	Four crowdsourcing platforms and a CS mailing list	How recruitment channels compare for recruiting participants with programming skills	Survey, quant.	613	2022

research purposes (unfortunately, we did not know this at the time of running our prior project [6]). Therefore, we do not suggest future researchers use developers' emails from software development platforms because, on top of breaching terms of services, it may tarnish researchers' reputation among the developers' community (particularly the open-source community). More broadly, we suggest researchers carefully read the terms and policies of platforms they intend to use to ensure they will not violate the terms.

3 EXPERIENCE WITH REVIEWERS

Number of Participants. Our primary publication venues have been CHI, SOUPS, and PETS. We received at least one reviewer comment about the number of participants in all of our submissions, suggesting that the community might not have a standard for the required number of participants for an interview or a survey study. While concepts like saturation for qualitative studies and power for quantitative studies exist, we believe that the community still lacks a review of standard practices and suggestions for best practices. Future research on the published papers with a specific focus on recruitment and the number of participants may provide insights into the standard and best practices of the community.

Participants' Levels of Experience. In all reviews, we also received comments about participants' experience. When conducting surveys or other remote research, we can rarely verify a participant's actual work status; and therefore rely on self-reported professions and years of experience in software development, among other demographic questions. We made the first effort to understand the differences between five recruitment channels to provide grounding for future research [5]. For example, we find that while participants may think they have programming skills, in particular from crowdsourcing platforms, they can still fail simple programming-related questions such as the appropriate value of a Boolean variable (True/False). Therefore, we strongly suggest future researchers use a screening survey on crowdsourcing platforms to ensure the required programming understanding of all participants.

Vague Meaning of Programming Experience. We note that the problem may not be participants misrepresenting their experience level. Instead, it may be that questions like "do you have programming skills?" (used by Prolific) or similar generic questions might have a range of participant interpretations. Participants who have only had one generic programming course or watched online videos may consider themselves as having programming skills. Researchers need to be aware of the type of developer experience they expect, and for work where participants need to do more complex tasks like test a library, a clear definition of "programming" is needed. We suggest designing a screening questionnaire rather than relying on general questions or self-reported demographics.

4 WHAT DOES THE COMMUNITY NEED?

Standardized Screening Questions. We believe a standard set of questions that can help researchers understand the level of experience developers have is needed to push forward the research in empirical software engineering. Such a set of questions may have programming, demographics, and other related topics that researchers can use to find a suitable sample for their studies. We do not think that all studies require professional, experienced developers. Some studies that test the usability of a prototype and compare it with prior technologies may well be tested with other groups of participants who may not be experts but can provide insights into the design process.

Collaboration Between Academia and Industry. A potentially sustainable approach to moving research forward is having software companies work alongside the research community in empirical software engineering. While a large technology company funded one author's Ph.D., they did not get support from the company to locate and recruit developers for studies. Perhaps explicit support from these companies in the contract and having a dedicated contact point for assisting Ph.D. students in recruiting developers may facilitate such studies. We believe that two of the primary beneficiaries of research from empirical software engineering are large software companies and the open-source community. Therefore, until they commit themselves to help researchers find participants and test new technologies with them, the research community may continue to find it challenging to move the field forward.

ACKNOWLEDGMENTS

We thank all of our collaborators in all publications that helped us gain these experiences.

REFERENCES

[1] Mohammad Tahaei, Alisa Frik, and Kami Vaniea. 2021. Deciding on Personalized Ads: Nudging Developers About User Privacy. In *SOUPS '21*. https://www.usenix.org/conference/soups2021/presentation/tahaei
[2] Mohammad Tahaei, Alisa Frik, and Kami Vaniea. 2021. Privacy Champions in Software Teams: Understanding Their Motivations, Strategies, and Challenges. In *CHI '21*. https://doi.org/10.1145/3411764.3445768
[3] Mohammad Tahaei, Adam Jenkins, Kami Vaniea, and Maria K. Wolters. 2020. "I Don't Know Too Much About It": On the Security Mindsets of Computer Science

Students. In *Socio-Technical Aspects in Security and Trust.* https://www.springer.com/book/9783030559571

[4] Mohammad Tahaei, Tianshi Li, and Kami Vaniea. 2022. Understanding Privacy-Related Advice on Stack Overflow. In *PETS '22.* https://doi.org/10.2478/popets-2022-0032

[5] Mohammad Tahaei and Kami Vaniea. 2022. Recruiting Participants With Programming Skills: A Comparison of Four Crowdsourcing Platforms and a CS Student Mailing List. In *CHI '22.* https://doi.org/10.1145/3491102.3501957

[6] Mohammad Tahaei, Kami Vaniea, Beznosov Konstantin, and Maria K. Wolters. 2021. Security Notifications in Static Analysis Tools: Developers' Attitudes, Comprehension, and Ability to Act on Them. In *CHI '21.* https://doi.org/10.1145/3411764.3445616

[7] Mohammad Tahaei, Kami Vaniea, and Naomi Saphra. 2020. Understanding Privacy-Related Questions on Stack Overflow. In *CHI '20.* https://doi.org/10.1145/3313831.3376768