POSTER: A Qualitative Study on Developers' Security Library Decisions

Lea Theresa Groeber*, Johanna Schrader*, Tamara Lopez†, Sascha Fahl**, Yasemin Acar*
*Leibniz University Hannover,
†The Open University, **Ruhr-University Bochum

***Abstract*—The recommendation to not reinvent the wheel and choose a library is common in programming, especially in the security field. Libraries are widely used to make the programmer's life easier. They provide solutions to common programming obstacles and thus can provide functionality and security features if applied correctly. Choosing a less-than-optimal library, however, may lead to problems ranging from poor usability to insecure code. In this paper, we investigate why developers choose a certain library with impact on security for their projects, which criteria are important to them, and which procedures they adhere to. If we can understand how they make their decisions, which resources they trust, which criteria they look for and what matters to them, we can better support informed and secure choices.**

As a first step, we conducted 20 in-depth interviews with professional software developers on how they choose libraries relevant to security. These interviews lead to several key findings: (1) Developers apply a "solution-oriented" search strategy where they quickly pick an early search result and engage with it, solidifying their choice as they learn the library (2) they care about a library being open source, usable, up-to-date and used by a large community (3) their choice is rarely limited by time-pressure, but often by their role not including security (4) they base their trust in libraries in the open source community and the assumption that an established, open source library will be secure. These and other findings unveil that software developers choose third party libraries with substantial trust in external developers, outsourcing product, company and users' security and privacy.

I. Introduction

We aimed to identify important factors impacting real-world security programming library selection, in order to answer the following research questions:

Q1 How and based on which factors do developers choose libraries relevant to security and privacy?
Q2 Which resources do they use?
Q3 Are there any factors that influence confidence in their choices?
Q4 How and why are their choices limited?

To this end, we conducted 20 semi-structured interviews with professional software developers who had chosen a library relevant to security or privacy in the recent past. In the interviews, we talked to them about their strategies and behaviours when making choices, allowing themes like security, privacy, trust in the libraries and their own choices, as well as limitations to their roles as developers to come up naturally.

We analyzed the 20 interviews using an inductive coding process, identifying several key findings, including:

- We learned that developers apply a "solution-oriented" approach, searching online and/or with the help of colleagues, basing their decision mostly on functional requirements of their project, existing knowledge or implementations, the usability of a library, its maintenance and its widespread use by a large community, which provides them with examples and online peer-created resources such as a large base of previously-answered Q&A.
- Developers place a large amount of trust in the open source community, both as a source of online help (in the form of Q&A) and as skilled reviewers of code, expecting open source software to be reviewed by experts for bugs and security. They also expect problems with popular open source libraries to have already come up and been solved in the past, such that they trust popular open source libraries' implementations.
- Contrary to common belief, developers rarely limit their research behaviour and the depth of their decision-making due to time pressure; however, their choices are often limited by understanding security and privacy to be tangential to their role as a software developer. This is true also in cases where they deal with critical consumer data, and in cases where they are solely responsible for their software.

Based on these findings, we widen the discourse for helping software developers to make more informed and safe decisions when choosing third party libraries with a relation to information security and privacy.

II. Results

A. Criteria for choosing a "good" library

We identified criteria that participants mentioned as markers of a high-quality library that they would be happy and likely to use. The most dominant criteria were that the library was open source, widely used, maintained and mature, as well as usable. All participants used open source libraries; the majority (14) had speficially chosen to do so because of the many benefits they perceived: They correlated open source libraries to reviews existing, the high chance that a library would be maintained or could be maintained by themselves or their company, the possibility to derive from community markers such as GitHub stars and forks how popular the library was, and the high chance to find questions about the library already answered online. Connectedly, participants were careful to

choose libraries that were mature: They looked for libraries with a large user-base (14 participants) and took care to only choose a library that was maintained (14 participants). For this, they often used open-source community features.

1) Usability: Usability was a crucial component to choosing and continuing to use a library, it was often mentioned by seven participants in the interviews before we prompted for it; altogether, 18 participants discussed usability. Participants said that they wanted a library to provide *"simplicity, with straightforward configurationns" (P09)*. If a library was "easy to handle (P16)", they were happy to use it. They reported previous bad experiences and felt that working with the library was *"strenuous" (P16)*. They would often end their search once they encountered a library that was sufficiently usable for their purposes, which means *"Spring does a lot for you, there's little left to configure" (P09)*.

However, bad usability led to some doubt in participants' ability to correctly use a library was intended.

Usable documentation was a crucial part or even standalone criterion for developers: ten participants mentioned that they looked for good documentation before even starting to engage with a library further. Documentation was considered as critical to task success and security.

B. Research behaviour

Every participant started their search for a library on the web first and possibly talked to colleagues later (8 participants). The most common resources on the web were StackOverflow (reported by 13 participants), GitHub (reported by 7 participants) and resources offered by the various library developers (7 participants). Table **??** illustrates which resources participants used while deciding for a new library.

Resources such as online forums, magazines, blogs and books were mentioned less prominently.

Participants generally classified their behaviour as "solution-oriented": they often based their decision on a past decision or prior knowledge, took re-usability into account, adhered to their search criteria as best as they could and rarely tried alternatives if their current choice satisfied their criteria.

1) Alternatives: Only five participants discussed engaging with alternatives before settling on their library. Fourteen participants explicitly said that they did not engage with or consider alternatives. This was caused by a perceived lack of alternatives, or simply the wish to reduce workload: *"We only tried this one library. Keeping the workload as low as possible." (C16)*

C. Limits to library decisions

The choices were often limited, and participants were not at all times satisfied with or completely trusting of their

2) Library decisions: Generally, participants liked to decide for a library that fulfilled their criteria; they wanted the library to fulfill functional requirements, be usable with as little time as possible, be well-documented and maintained, hopefully by someone else. They A common theme mentioned was re-use: When choosing a library, their choice was often influenced by previous choices made within their company or for their project. A library that was already in use or previously known, such that that either they personally or their organization as a policy wanted to use.

choices. We had anticipated time-pressure and lack of security education to be factors. However, we found that the reasons for making decisions with less depth than maybe desired (both in strategy and in outcome) were multifaceted.

1) Time-Pressure: Only seven participants mentioned time pressure as a factor on choosing libraries faster than they ideally would want to.

P16 stated that his team is agile with two-week sprints, such that there is not a lot of time to choose a library.

Eleven participants explicitly said that, especially in security-critical cases, time pressure does not play a role in choosing a library. One participant paraphrased: *"For us, it's not like we don't have the time and have to compromise when choosing a library ... there's not a lot of pressure. (P018)*.

P19 reported that time-pressure does not play a role in his company, especially in security-critical cases: per company-policy, developers are expected to prioritize security over quick results.

2) Self-Assessment: Participants mostly implicitly or explicitly expressed that their role did not include security. This held even when they were the only software developer working on software, and also when user data were at play.

One participant (P09) said that other, more security-expert developers would be likely to review open source libraries, however, they themselves did not contribute to open source reviews (security or not), citing high complexity and lack of time.

3) Trust in open source community: Participants generally stopped their search because they trusted in their early choice. This trust was mainly placed in the open source community, as mentioned as a criterion by 14 developers; when elaborating, 18 altogether stated their trust in the open source community. Generally, acceptance by others was considered a proxy for a "good" choice: One developer stated *"I assume that the developer community, or, the greater community that uses the module, monitor this for me." (P05)*, another said *"I only fleetingly looked at the source code, but it's used widely and by many companies and people, so I have a high level of trust." (P02)*. However, no interviewee mentioned that they had ever contributed to open source projects, reviewed open source code, written a review or answered online questions themselves. It is possible that they did not mention they because it did not come up in the interviews; however, we feel that there is a tendency to passively rely on a community for which it is unclear how many active contributors are actually actively contributing. This goes in line with only four participants explicitly mentioning looking for audits, code reviews or certifications.

Maturity, *"a library that has been on the market for a while" (P19)*, was another theme that came up when discussing trust in a library. Participants felt that the continued use and engagement of other developers with a library was a good indicator of its security and functional usefulness.