TABLE 2. ABRIDGED VERSION OF THE SEMI-STRUCTURED INTERVIEW GUIDE

Section A: Participants, Projects, and Organizations
To get started, can you tell us about the type or domain of software you primarily develop?
How would you describe your target client for software? Is it general people, government, or other software firms?
How did you get to learn about security? Does your company arrange training/workshops for you? Or self-learning?
What is important in terms of security in terms of your product/software?
What potential threats do you consider that may compromise the security of system?
Are these the threat assumptions you normally consider in the domain you work on/at work?
Section B: Organization and Security
Do you remember being constrained by any factors, such as Deadline/Time, Requested Features, Dependencies, or others, when programming that may have affected/compromised security guarantees?
How would you describe the software development process you follow?
Do you remember your organization's existing coding standards, national regulations or any other software development process aspects having any influence on security guarantees?
Have you implemented security functionalities through in-house development instead of relying on third-party libraries? What situation necessiated this?
Between using third-party libraries for security-sensitive functionalities and implementing security-specific features on your own, which one do you prefer?
Can you tell us more about the team you work with?
Can you tell us about your team structure and security specific functions/components?
Do you write test cases specifically for covering/testing security-related requirements? Can you give us an example?
What are the consequences if the security requirements in your software are not met?
Section C: Organizational Context of SAST
Why do you favor using <SAST (s)> in your organization? What events led to this decision?
Where does the SAST come in the Software Development Life Cycle (SDLC) you follow? Can you walk us through the process?
Do all (security-related) team members know/receive training about the SASTs that you use?
Are there generally any SAST-specific requirements from the user/customer?
Can you please walk us through the process of selecting such a security focused tool?
How are SASTs Security helpful for Agile/Scrum processes?
Can you tell us more about the events which influenced you in becoming a SAST user instead of focusing on being a manual technique based user?
How much is generally the cost in dollar value for licensing and/or using SASTs?
You have mentioned that your organization relies more on the SASTs over Manual Techniques (or vice versa). Why is that?
How do you generally handle security bugs in product?
Section D: Expectations from SAST
Consider the following statement: *"When using an automated code review/scanner, a static tool should be capable of reporting all the issues in the code as far as static analysis allows"*. What is your opinion regarding this statement?
"When using an automated code review/scanner, a static tool should only show results it is 100% certain about, even if it means it may miss a few potential issues", What is your opinion regarding this statement?
Depending on the difficulty/nature of vulnerable code issues, some issues might be more difficult than others to detect by tools. Therefore, would you consider tools that are not perfect (*i.e.,* may miss some vulnerabilities) to still be acceptable to use? What is your opinion regarding this?
You mentioned that you prefer FN/FP over FN/FP; Do you think this is purely because of the kind of software you work on, or do you think it is shared in the general developer community or in <type> developer community?
How would you describe your overall impression when using security analysis tools for analyzing custom implemented security features?
Does the tool clearly present detected security vulnerabilities, provide any explanations, link detected security vulnerabilities to known examples? Anything else you prefer these tools should have/report that is currently not available?
Section E: Impact of Unsound/Flawed SAST
Have you ever been in a situation where there was a vulnerability in your software, which should've been detected by a SAST but was not? How did you handle it?
If in case your software has a security issue which was not found due to buggy SAST, how do you handle the consequences?
"Just because a tool report states that there are no security errors does not mean the software is secure, since the tool itself may be buggy". Can you please elaborate on your opinion regarding this statement?
Do you expect the SAST to catch everything?
What happens if you find something that a SAST should catch, but does not? Do you report it to the SAST developers?
Have you encountered any situation where any developer tried to evade SAST security checks by abusing flaws?
If you ever reported a problem to SAST developers, did you ever get a response and a follow-up fix to the issue that you reported (with example)?
(Previous context) What role do you think fuzzing tools play in comparison to SASTs here? Do you think fuzzing tools can replace SASTs?
Section F: Challenges and Improvements
Have you ever considered designing and using an in-house SAST? What limitations of existing tools motivated you to do so?
If you were given unlimited resources to fix/create the perfect SAST, what issue would you address before anything else?
Do you have any kind of final thoughts or anything that you would like to follow up on?