Lars Grunske, Janet Siegmund, Andreas Vogelsang (Hrsg.): SE 2022,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2022 61

IDE Support for Cloud-Based Static Analyses

Linghui Luo[1], Eric Bodden[2]

Abstract: We present a user study with developers at Amazon Web Services on their expectations of IDE support for cloud-based static analyses. The paper was originally presented at ESEC/FSE 2021. Many companies are providing Static Application Security Testing (SAST) tools as a service. These tools fit well into CI/CD, because CI/CD allows time for deep static analyses on large code bases and prevents vulnerabilities in the early stages of the development lifecycle. In CI/CD, the SAST tools usually run in the cloud and provide findings via a web interface. Recent studies show that developers prefer seeing the findings of these tools directly in their IDEs. Most tools with IDE integration run lightweight static analyses and can give feedback at coding time, but SAST tools take longer to run and usually are not able to do so. Can developers interact directly with a cloud-based SAST tool that is typically used in CI/CD through their IDE? We conducted a user study to explore how such IDE support should be designed. Through this study we identified the key design elements expected by developers and investigated whether an IDE solution fits better into developers' workflow in comparison to a web-based solution.

Keywords: IDE integration; Static analysis; Cloud service; SAST tools; Security testing

1 Summary

More and more software companies are integrating Static Application Security Testing (SAST) tools into their continuous integration (CI) or continuous delivery (CD) pipelines. Popular SAST tools are often cloud-based and offer hooks to integrate with CI/CD systems such as GitHub Actions, Jenkins and Travis CI. The common workflow of using such tools is as follows: developers write code in their IDEs, and then commit and push code into a Git repository, which will trigger a SAST tool to run an analysis on the code. To view the analysis result, developers need to login to a dashboard in their web browsers. Although such workflow is widely spread, it poses a usability issue for developers—they are taken outside their IDEs, where they write code. Recent studies report that developers want to see the result of these SAST tools directly in their IDEs [DWA20, CB16]. Our recent study [Lu21] asks the question how we can give developers access to cloud-based SAST tools directly through their IDEs, and if this improves developers' workflows. To answer these questions, we conducted a multiple-staged user study with software developers at Amazon Web Services.

First, we started by interviewing nine developers to understand their expectations of how cloud-based analyses should be triggered from an IDE, how analysis results should be

[1] Universität Paderborn, Germany linghui.luo@upb.de
[2] Universität Paderborn and Fraunhofer IEM, Germany eric.bodden@upb.de

displayed there, and what UX features they would like. While some participants expected to trigger the analysis manually by clicking a button in the IDE or when they build the project, the others would like the analysis to be automatically triggered. All participants told us they expected the IDE support to display the result automatically once the analysis is completed in the cloud. Which part of the result should be displayed in the IDE depends on developers' primary goals. If they want to improve the overall code quality, showing the entire project is desired. If they are implementing a new feature, only findings that are in the diffs should be displayed.

Guided by our findings from the interviews, we developed an IDE prototype for an existing cloud-based SAST tool—Amazon CodeGuru Reviewer [AW19], using its infrastructure for CI/CD. Users still need to push their code to a remote Git repository. Yet the IDE prototype allows users to interact with the cloud service directly inside their IDEs, i.e., they can request CodeGuru Reviewer to reanalyze the code and view the findings along with the code.

Once we had the IDE prototype, we presented it to the same group of developers we interviewed before to evaluate whether the design met their expectations. While these developers were satisfied with most features built in the prototype, they found existing mechanisms for CI/CD, e.g., code uploading via Git, were cumbersome in the IDE.

Finally, to test if the IDE solution was an improvement over the web-based solution, we conducted a within-subjects usability test with 32 developers. We found that using the IDE prototype developers performed code scans three times more often than using the web-based solution. Our measurements also show a promising reduction in time for fixing code. However, bringing the findings of the tool into the IDE did not necessarily improve developers' workflow as we learned from the after-session interviews. Specifically, they expected real-time feedback, quick validation of each fix, more seamless analysis of code and interactive ways to suggest rescan. We also noticed that some developers had limited understanding of the capabilities of SAST tools and confusions regarding the built-in workflow of such IDE integration. Future work should study how to make such IDE integration more intuitive for users.

2 Data Availability

The study was conducted during the first author's internship at Amazon Web Services. Due to company policies, the interview transcripts and IDE prototype are not available. The interview questions, the list of codes for analyzing the interview data, the tasks used in the usability test and survey questions are available at `https://github.com/linghuiluo/FSE21Study`.

Bibliography

[AW19] Amazon CodeGuru Reviewer, `https://aws.amazon.com/codeguru`.

[CB16] Christakis, Maria; Bird, Christian: What Developers Want and Need from Program Analysis: An Empirical Study. In: Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering. ASE 2016, Association for Computing Machinery, New York, NY, USA, p. 332–343, 2016.

[DWA20] Do, Lisa Nguyen Quang; Wright, James; Ali, Karim: Why do software developers use static analysis tools? a user-centered study of developer needs and motivations. IEEE Transactions on Software Engineering, 2020.

[Lu21] Luo, Linghui; Schäf, Martin; Sanchez, Daniel; Bodden, Eric: IDE support for cloud-based static analyses. In (Spinellis, Diomidis; Gousios, Georgios; Chechik, Marsha; Penta, Massimiliano Di, eds): ESEC/FSE '21: 29th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, Athens, Greece, August 23-28, 2021. ACM, pp. 1178–1189, 2021.