Hochschule
Augsburg University of
Applied Sciences

Tools for Security Testing in Continuous Integration Pipelines

Michael Wager
Faculty of Computer Science
University of Applied Sciences Augsburg
Augsburg, Germany
mail@mwager.de

***Abstract*—Nunc viverra nibh at magna vulputate, sit amet accumsan purus elementum. Aenean rhoncus augue ac pretium pretium. Integer facilisis vestibulum porta. Curabitur nisl ligula, sollicitudin non eleifend sed, accumsan interdum mi. Nullam semper vestibulum eros vitae tempor. Vestibulum magna ipsum, auctor ut sem vitae, imperdiet egestas nibh. Sed ante nunc, blandit id tortor sit amet, consectetur cursus purus. Sed id velit laoreet, accumsan mi non, congue nulla. Morbi magna libero, vestibulum quis ipsum a, malesuada pulvinar eros. Mauris vulputate, nulla vitae lacinia efficitur, purus augue sollicitudin ante, non dapibus est massa a tortor.**

I. Introduction

Software development is still a very young craft. But we are writing the year 2022 and the amount of devices containing more and more complex software is growing rapidly. Comprehensive quality assurance, especially in the field of web application development, also could deserve a lot more attention. Nevertheless, while a few years ago it was quite common practice to edit files on a production server directly via FTP in order to deploy a fix or a change, today methods and processes to assure a certain level of quality are used quite often. One of that method is so called Continuous Integration (CI). The idea is to provide immediate feedback to the developers directly after the integration of changes to a codebase. On every push to the repository a so called CI server starts executing a pipeline of predefined tasks to ensure high quality and to make sure the changes do not break existing functionality. Common tasks are static code analysis to ensure a clean coding style or to catch simple development mistakes or unit testing to make sure the codebase maintains a high quality standard.

As this is also still a very young process, it is pretty obvious that software security also does not play a big role. But as the growing interconnectivity through the pandemic and also recent attacks against critical infrastructure moving through the media, cyber security gets a greater attention in recent times. In the context of the Security Development Lifecycle [1], one idea to ensure a great level of application security is to automate as much as possible directly during the CI pipeline.

This work give an overview of the current state of continous security testing, its categories, possibilities and limits and will also provide an overview of existing tools.

II. Related work

–¿ halt checken was da ggf fehlt, bissl drauf eingehen und zusammenfassen

- Check andere Papers -¿ SO: Was macht anderes Paper und was mach ich hier anders? nur 1-3 Sätze pro Paper! -¿ auch: von wann sind die papers? Sind die aktuell? Ist das noch relevant und/oder was hat sich geändert? -¿ nur die die ziemlich ähnlich sind? -¿ Ähnliches Ziel. So wie alle die ich bis jetzt gelesen hab :D

III. Background

- Was muss leser minimum verstehen um Arbeit zu verstehen. Also die ”Fachliche Grundlage” für mein Paper - Grundlage legen: was is CI, was is DevOps ===¿ UND: was is DevSecOps!!!!!!!! Thema Security!!! - hier vielleicht papers zum thema devOps referenzieren: zb Agile Manifesto, Wasserfall, ETC! - WICHTIG VORHER: READ STUFF! und direkt daraus bullet points machen

IV. Tools

Hauptaufgabe des Papers: Ordnung in Chaos

- Kategorisierung hier! dann pro Kat ein paar Tools - auch auf CIs eingehen? Jenkins, GitLab etc

A. Tool 1

Sed nisl eros, semper sollicitudin

B. Tool 2

Sed nisl eros, semper sollicitudin

C. Tool N

Sed nisl eros, semper sollicitudin

V. Discussion

Kapitel eher nennen Discussion oder Evaluation

- und wird analysiert was so vorher zusammengetragen wurde! - Eigenschaften und Vergleiche der Tools
- zB Languages -¿ unter zb diesem Punkt könnte man dann evaluieren. Zb 90prozent JavaScript
OSS gibts nix für Go, bla bla
- Faktor Mensch: Es gibt studien mit Interviews, was sind Knackpunkte, wie reagieren Entwickler, weil Zeitgründe, viele Warnungen etc. Was kann man hier tun.?
- Was hab ich denn sehen nach meiner Tool recherche!?????????
- EMbedded / WebDev / Desktop dev etc?!?!?
- Wie gehts weiter mit embed, Industrial Bereich? etc......
- hier mussen paar Fragen entstehen die ich dann in Conclusion beantworten kann!!!!!!!!!

VI. Conclusion

- Fragen aus Discussion nun beantworten!

References

[1] M. Howard and S. Lipner, “The security development lifecycle,” 2006.