**Tools for Enforcing Secure Code**

Benjamin M. Brandhorst

University of Maryland Global Campus

SDEV 360 6380 Secure Software Engineering

Professor Kevin Woodson

February 23rd, 2020

**Survey of automated tools for enforcing secure code development**

When researching Static Application Security Testing (SAST) tools for this week’s assignment, I quickly came to the realization that there are far too many for me to list individually. The Open Web Application Security Project (OWASP) site alone lists over sixty tools ("Source code analysis tools," n.d.). In order to narrow things down a bit, I selected four applications to survey. I will summarize the strengths and weaknesses of each of these programs and reveal which I picked to analyze source code with. The applications included in this survey are Coverity, Fortify Static Code Analyzer, Klocwork, and Codacy.

**Coverity**

Coverity is a proprietary SAST tool offered by Synopsys which supports 20 programming languages. This tool is one of the most popular of its class and is widely used by software development teams. I was unable to find a published pricing structure of the tool because the Synopsys website prompts interested parties to provide contact information so their sales team can reach out to them. However, looking at reviews around the internet, a common complaint is how expensive the tool is for development teams to use. Outside of the cost though, reviews are positive and mention the low false positive rate and ease of use as being contributing factors in selecting this tool over the competition ("Coverity static code analysis," n.d.).

**Fortify Static Code Analyzer**

Fortify Static Code Analyzer is a product and service offered by Micro Focus that is available for more than 26 programming languages. Much like Coverity, this is a popular SAST tool which seems to position itself as a lower cost alternative to Coverity. I could not find any

current pricing but digging through older reviews of the tool showed that in 2007 the cost was $1,200 per developer ("Fortify software source code analysis product review," 2007). This same review indicates that the pricing structure is on the lower end of prices for similar offerings. The strengths of this tool include the wide variety of languages supported. It can also integrate with the Eclipse or Visual Studio IDE to give real time feedback to developers. However, some reviews indicate the interface is not easy to navigate and may be confusing for those who are not full-time developers ("Fortify software source code analysis product review," 2007).

**Klocwork**

Klocwork is another tool for SAST. This one is offered by Rogue Wave Software and is used by development teams and individual developers. This tool is more focused than either Coverity or Fortify Static Code Analysis; meaning that C, C++, C#, and Java are the only supported programming languages. However, as long as your development team is using one or more of the supported languages, this tool is highly rated. Klocwork has an IDE plugin that offers real time feedback during development. Reviews of the tool indicate that, if properly setup, it is very effective at detecting actual bugs and vulnerabilities and returns a low number of false positives. However, users also say that the command line interface is less user friendly than a Graphical User Interface (GUI) would be. Additionally, the results of scans cannot be easily imported into an excel file, which makes analyzing scan results more cumbersome than need be ("Klocwork," n.d.).

**Codacy**

The final SAST tool and the one I selected to analyze source code with is Codacy. Founded in 2012 and based out of Portugal, Codacy is a relatively young tech startup that

provides services for over 1,000 software development teams. They offer both cloud and enterprise hosting, with cloud services starting at 15 dollars a month per user. The tool can analyze 28 different programming languages and uses an intuitive GUI to categorize and display a listing of issues found in the scan of the source code. Users praise the software for its ease of setup and customization. Additionally, Codacy includes a “per-commit strategy” which triggers an analysis of a specified repository for each commit. It also analyzes past commits so development teams can see trends and the evolution of the quality and number of created and deleted issues (Leroy, 2017). However, users also criticize the software for its inconsistent documentation and the cost of the enterprise solution. Additionally, some reviewers had issues with frequent repository disconnects when the tool was not in use ("Codacy," n.d.).

**Analyze code using tool**

An open source project from the website [freeproject24.com](http://freeproject24.com/) called “Online Car Sales System” was selected to be analyzed. This project has approximately 1600 lines of source code and was uploaded to a private [Github](http://www.github.com/) repository so it could be used for this demonstration.

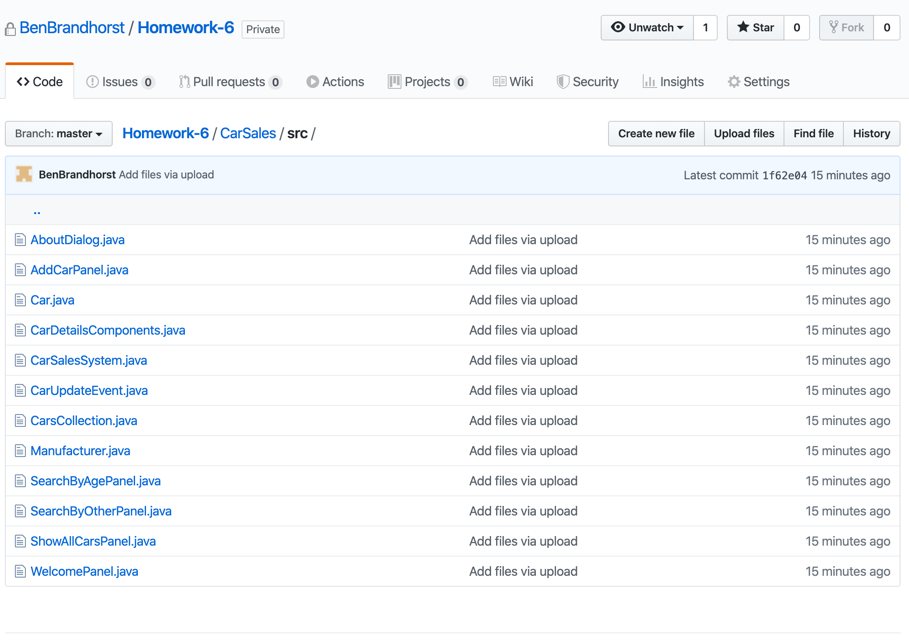


Figure 1 – Files located within Github repository

From there, the cloud based Codacy application was used to import and scan the contents of the Github repository. The application includes a dashboard that allows users to track several different repositories at once and see the number of identified issues discovered at a glance.

A screenshot of a cell phone

Description automatically generated

Figure 2 – Codacy dashboard example

Clicking on the highlighted repository will give more detailed information, including a quality evolution and a categorization of the identified issues.

A screenshot of a computer

Description automatically generated

Figure 3 – Codacy repository details

Additionally, selecting “See all issues” gives a detailed breakdown of the 34 identified issues with the scanned source code. This breakdown displays each line of problematic code and provides a short explanation of why it poses an issue.

A screenshot of a cell phone

Description automatically generated

Figure 4 – Codacy Current Issues page

Finally, using the “All patterns” dropdown menu sorts the identified issues into categories. Here we can see that 12 of the 34 issues are directly related to this project not using packages for the class files it contains. An additional 6 issues stem from empty method bodies being used.

A screenshot of a social media post

Description automatically generated

Figure 5 – Codacy All patterns drop down menu

Of the different tools experimented with, Codacy was the easiest to setup and get running on a small project. That said, it seems as if this category of tool is a very competitive space and I am sure that any number of other proprietary or open source solutions could be used with a similar effect. However, there is something to be said for how little time had to been spent setting things up and how quickly the Codacy tool can be integrated into new or existing projects.

References

Codacy. (n.d.). Retrieved from https://www.capterra.com/p/144689/Codacy/#reviews

Coverity static code analysis. (n.d.). Retrieved from https://www.capterra.com/p/163552/Coverity-Static-Code-Analysis/

Fortify software source code analysis product review. (2007, August 15). Retrieved from https://www.scmagazine.com/review/fortify-software-source-code-analysis/

Klocwork. (n.d.). Retrieved from https://www.capterra.com/p/136486/Klocwork/

Leroy, S. (2017, July 10). Codacy: A new easy-to-use code quality and review automation solution - DZone DevOps. Retrieved from https://dzone.com/articles/codacy-a-new-easy-to-use-code-quality-and-review-a

Source code analysis tools. (n.d.). Retrieved from https://owasp.org/www-community/Source\_Code\_Analysis\_Tools