**Discussion Topics: Application Security**

Select one of the following and find at least two (2) articles on the topic. In your post include links to your sources, an explanation of the topic, when it should be used, why it should be used, and what tools are available, if any to help with implementation.

1. Static Analysis
2. Dynamic Analysis
3. Dependency Scanning
4. Source Code Integrity and Code Signing

***Before you submit your thread, put your name in the subject line.***

For this discussion post, I will be covering static analysis. Throughout my search, I found an article about Static Analysis on TechTarget by Alexander S. Gillis, and another by Stuart Foster on the Perforce website.

Static analysis or static code analysis debugs a program without needing to execute it first (Gillis, 2020). Debugging this way helps provide a better understanding of the code and compares code to industry standards (Gillis, 2020). Static analysis is typically handed through software developers and "quality assurance teams" (Gillis, 2020). There are different types of static analysis, including "control analysis, data analysis, fault/failure analysis, and interface analysis" (Foster, 2023).

The best time to utilize static analysis is when developers need to find "programming errors, coding standard violations, undefined values, syntax violations, and security vulnerabilities" (Gillis, 2020). Static analysis is used in the early stages of the software development life cycle, before software testing begins (Foster, 2023).

There are times when static analysis should be used either in addition to other debugging techniques or by itself. Some of these scenarios include wanting to save developer time and address issues early on, when wanting to test every code execution path, and when looking for better accuracy (Foster, 2023).

There are many tools available for static analysis. Before choosing a static analysis tool, knowing what programming language you are using beforehand and understanding coding standards helps make the right choice (Foster, 2023). Static analysis tools help automate testing processes, ensuring application reliability, security, and maintainability (Foster, 2023). A few of those tools include Embold, Kiuwa, and Pycharm (Gillis, 2020).

**References**

Foster, S. (2023, June 7). *What Is Static Analysis? And What Is Static Code Analysis?* Perforce Software. https://www.perforce.com/blog/sca/what-static-analysis

Gillis, A. (2020, July 31). *What is Static Analysis (Static Code Analysis)?* WhatIs.com. https://www.techtarget.com/whatis/definition/static-analysis-static-code-analysis

**Assignment Requirements and Grading:**

1. An initial post of approximately 250 words is due by **Thursday, 11:59 p.m., CST**.
2. For the initial post to be considered substantive, it should be at least 250 words in length and fully cover the topics being presented. Single-sentence definitions or responses will not be awarded points.
3. Submit your post by clicking on the **Assignment Link** above, then **Create Thread**. You must create a thread in order to view your peers' posts. Tip: Create your post in a Word document and then copy and paste your work into the thread.
4. A minimum of three (3) responses, **to the original threads of other students**, of 100-200 words each are due by **Sunday, 11:59 p.m., CST**.
5. To view the rubric grading criteria, click on the following link: [Discussion Board Grading Rubric](https://content.bellevue.edu/cst/csd/rubricdbv3.pdf).

**(50 points)**

Hi, Brett! I think you did a great job on your post for this module! You included links to your sources and explained dependency scanning, when it should be used, why it should be used, and what tools are available. OWASP is most definitely a great tool to implement, and the fact that it is open source makes it even better! It is great when the software development community comes together to release helpful tools and sites for all to use. I completely agree that organizations should perform scans before development. It will help identify issues before they are released to the public.

Hi, Adrian! I enjoyed reading your post on this discussion. It was very thoughtful, and the information you provided from your sources expanded on the knowledge I gained from researching the topic. I agree that static analysis should be done in the early stages of the software development lifecycle. Catching errors in a program before it is deployed saves time and resources, plus provides many more benefits. I like how you included how static analysis supports compliance regulations. That can be extremely important. Some other tools that I found are Embold, Kiuwa, and PyCharm. Out of those three, I have used PyCharm before and have had a decent experience with it.

Hey there, Megan! You did a fantastic job of defining how dependency scanning works, when it should be implemented, why it should be used, the tools available, and best practices. Based on what I have learned about these different types of application security, I see how Static Analysis, Dynamic Analysis, Dependency Scanning, and Source Code Integrity and Code Signing all have huge advantages and understand why they should be used. When developing for a company or team, it makes me curious which application security practices will be implemented and which will not be. As great as implementing them all would be, I guess it is usually unrealistic.