Topic 4 Discussion 1

What are the two types of source code analysis available to testers? Provide a definition and brief summary of each. Which would you choose to use to identify a flaw and why? How would a flaw in source code become a vulnerability? Provide an example.

Hello Class,

In the context of software testing, there are two primary types of source code analysis: static code analysis and dynamic code analysis. Each serves a distinct purpose in identifying potential flaws and vulnerabilities within software systems.

Static Code Analysis

This method involves examining the source code without executing it. It focuses on identifying potential vulnerabilities, coding standards violations, and other issues by analyzing the code structure, syntax, and patterns. Tools used for static analysis can detect problems such as buffer overflows, SQL injection vulnerabilities, and other security flaws early in the development process(Dewhurst, 2013). This approach is beneficial because it allows developers to catch issues before the code is run, making it easier to address them without the complications that arise during execution.

Dynamic Code Analysis

Dynamic code analysis involves executing the code in a runtime environment to identify issues that may not be apparent through static analysis(Bhattacharjee, 2024). This method tests the software's behavior under various conditions, allowing testers to observe how the application interacts with its environment and to identify runtime errors, memory leaks, and security vulnerabilities that occur during execution. Dynamic analysis is particularly useful for identifying issues related to performance and security that only manifest when the code is running.

Choosing an Analysis Method

If I were to choose a method to identify a flaw, I would opt for static code analysis. This choice is based on its ability to catch vulnerabilities early in the development cycle, which can save time and resources in the long run. By identifying issues before the code is executed, developers can address them proactively, reducing the risk of security breaches and improving overall code quality.

From Flaw to Vulnerability

A flaw in source code becomes a vulnerability when it can be exploited by an attacker to compromise the system's security. For example, consider a flaw where user input is not properly sanitized before being processed by the application. This oversight can lead to a SQL injection vulnerability, where an attacker can manipulate the input to execute arbitrary SQL commands on the database. Such vulnerabilities can result in unauthorized access to sensitive data, data corruption, or even complete system compromise(Kuiwan, 2023).

References:

Bhattacharjee, S. (2024, September 4). *Static vs. dynamic code analysis: A comprehensive guide*. VFunction. https://vfunction.com/blog/static-vs-dynamic-code-analysis/

Dewhurst, R. (2013). *Static Code Analysis | OWASP*. Owasp.org. https://owasp.org/www-community/controls/Static\_Code\_Analysis

Kiuwan. (2023, December 7). *What Are Code Vulnerabilities? - Kiuwan*. Www.kiuwan.com. https://www.kiuwan.com/blog/what-are-code-vulnerabilities/