Jordan Colbert

CS 405

October 24, 2021

Southern New Hampshire University

Journal: Portfolio Reflection

Adopting a secure coding standard is integral to maintaining security in the software world. Threats can only be stopped when programmers know what they are fighting and how to do so. Utilizing techniques to prevent SQL injection or overflow by running assertions or exceptions can go so far as to stop the program from running. This is one of the earliest methods in defense-in-depth, as the programmer is the first line of defense in this case. Leaving security protocols and decisions to the end can cause harm as well. By the time a program has been finished, if a secure coding standard has been established, the code may need to be reworked to support the expectations. For example, if security is ignored initially, a developer may not try to catch “OR \*” statements for SQL injection. Depending on the program, this may be a tricky issue to revise.

Risk and benefit cost analysis is another important facet of security, as it governs what will be implemented. Ultimately, an organization must analyze what they are seeking to accomplish with a program first. Once that is established, security measures can be weighed against each other. Certain protocols may work better in certain situations. Some programs are far lower in scope—a calculator app may not need encryption of any kind, while a database with multi-factor authentication will absolutely need it. Cost and manpower are other ways to measure what is feasible. After all, it is impossible to create a program that is completely secure. There will always be exploits that arise and budgets are finite. A company should learn how to balance needs and wants when analyzing a project overall.

A policy of zero trust is ideal in virtually all security policies. This is because there is no real way to know if someone is a malicious user or not without steps in place. Assuming users have no bad intentions will immediately result in a compromised program. In this sense, allowing access after and only after verification makes sense. These various defenses and preparations make it clear why security policies should be executed and enforced. However, it is important for the policy to be updatable as time goes on. New recommendations will appear as new threats appear. Tenets of the security policy could change with the times. However, this should not be interpreted as a reason to not follow the policy. It only refers to the policy supporting improvements. Keeping these considerations in mind, security policies are ideal, especially ones that can be built upon.