Jason Restucci

CS-405

Reflection

Security should be incorporated into every step of development. It should be baked into the design of the system, adding layers of security as development progresses in a “Development in Depth” approach. Adding security at the end of the project will only result in never-ending problems and complete rewrites of systems. Developers should also adopt a secure coding standard to produce more secure and more consistent code throughout the project.

Knowing possible risks to your system, evaluating them and the cost-benefit of mitigation is important as well. This involves evaluating the likelihood that the risk could happen and how severe it is. Things that are more likely and more severe take priority. Additionally, it is important to analyze the risk-to-cost ratios of potential mitigation strategies to choose the best mitigation strategy possible. Some risks might also be easily preventable, while some might constitute an additional layer of security. Having a thorough analysis and list of these will affect your security and financial decisions moving forward. A good strategy to protect against the many attack vectors your project might face is to adopt a “zero trust” policy. This means that no device or user is automatically trusted. Absolutely anything and everything coming in or going out of the system must be verified, authenticated, and authorized.

When developing a security policy you should make sure you have the following things: A DevSecOps pipeline, a list of principles and a list of standards, policies for encryption, as well as policies for triple-A (Authentication, Authorization, and Accounting), and finally, proper procedures for testing (unit testing, penetration testing, etc…). Furthermore, you should train your developers as well as perform regular security audits on the policy and the employees as well. This ensures your policy is up to date and your developers do not get too relaxed with the policies.