Wesley Blackwell

Module Eight Journal Reflection

Over the course of this course, we have learned about the purpose of security, how it should be implemented, and how to ensure that our system is as secure as possible. The adoption of a secure coding standard mitigates vulnerabilities, and potential vulnerabilities, in a system. This standard helps developers work toward a product that has security built in at every stage of development. The main purpose of this is to ensure the system has the best possible security while reducing the cost and maintaining the functionality of the system.

We should never leave security until the end. This practice is detrimental to projects and development. Designing a functional system is great, but when we have to practically dismantle the whole architecture to implement security at the end, we create more problems. Problems like overlooked vulnerabilities, loss of functionality, loss of timeline, and heavy cost. Other issues will almost certainly arise with updating, operation, and implementation. Developers should always avoid leaving security until the end of a project because preventative development of a system is ideal when it comes to security. This practice leads to a sturdy, robust system with a well layered Defense in Depth architecture.

Typical business needs to develop products, including software products, are risk assessments and cost/benefit analysis. When we talk about security, the cost of implementing it at the end can be astronomical. Using a risk assessment, we can identify areas of vulnerability and rate their risk levels based on priority. Once this assessment is completed, we can then perform an effective cost/benefit analysis of the project to identify the cost of each individual phase of the project. This gives the business a clearer price tag and timeline.

Zero trust refers to systems having a default setting of automatic distrust of everything. This means that at every instance each system attempting to access another system, authentication must occur no matter what. The standard approach says trust but verify, meaning not to default to distrusting but, verify access. This approach ties in directly with the AAA framework. Zero trust essentially means nothing is safe and we cannot trust anything. Always verify with authentication for every system, every time. This attitude is naturally preventative by authenticating every access attempt, inside and outside the system.

DevSecOps allows us to adopt security standards and practices. Development of the security policy and recommendations helps bridge the gap between all of these concepts. Using our security policy we can implement all aspects of secure coding in some form throughout the development process.