Brian Engel

Module 8 Journal

Adoption of a secure coding standard is essential to security and protecting an organization’s data and its customers personal data. To do this security cannot be left until the end. It should be incorporated in every step of the process. There are many ways to secure an application such as defense in depth, which creates overlapping layers that will catch any threats that make it through the layers above. You can use encryption for data at rest, in transit, and in use. Triple-A or authentication, authorization, and accounting is important to make sure the user is who they say they are, give them access to only what they need to get their job done, and create a paper trail in case something happens. A secure coding policy is created by making rules on how to handle specific situations in coding that can lead to vulnerabilities. This way the entire team is aware and knows how to handle the situation in the same way.

If a vulnerability is found the risk should be evaluated and ranked by priority in which it should be fixed. The priority should be a combination of how likely the vulnerability is to occur and how much damage it can do. Just because a vulnerability has a low priority does not mean it should not be fixed. While no system can be 100% secure, vulnerabilities should be fixed as they are discovered.

Zero trust is a policy that doesn’t trust any sources, both external and internal. It uses a proxy and gateway to validate user credentials and controls the flow of the user’s movements within the system. It is a change from the castle type system where once you are in you are free to access anything within the walls to a more granular system where every step is checked for authentication and authorization, meaning that lateral moves can’t be made and will always be checked for authorization levels adhering to the principle of least privilege easier.

To implement a security policy there are several steps that should be taken. First, security should be included in the planning process. Once secure code is written and the system has been though code reviews for logic errors, automation should be implemented in the form of unit tests, static analysis, and dynamic testing to discover unseen errors. Once implemented, automation can also be used for logging and monitoring user actions. Finally, constant improvement is required since vulnerabilities will inevitably be found and technology is constantly changing.