The adoption of a secure coding standard is a necessity for most applications. While an initial assessment can make it appear that implementing security from the start of a project can increase costs and time, it is significantly more expensive and time consuming to retroactively implement security. Furthermore, when there is an attempt to implement security after initial development, there is a higher chance for it to be less comprehensive (to miss a security vulnerability). By adopting a secure coding standard before a project, it becomes easier to implement and ensure that all (or most) known vulnerabilities are covered throughout development.

When it comes to determining the level of security to implement, there needs to be an assessment of the risks, costs, and benefits. There are a large number of security vulnerabilities to cover, and some are easier and less expensive to mitigate than others. The evaluation and assessment of risk and cost benefit of mitigation needs to be performed to determine priorities for mitigation. This can help make the secure coding standard more effective by implementing standards which are more cost effective based on the risk, and developers do not waste as much time attempting to resolve vulnerabilities with lower risks and have lower benefits compared to the costs.

Zero trust is an interesting security framework that changes how security is implemented for a system. Traditional systems are secured as a whole, trusting anything inside the system and not trusting anything outside the system. However, if a malicious user gets into the system, they would gain access to much of the data and system resources as they are now inside the trust barrier. Zero trust helps address this by implementing security barriers around individual components inside the system. This makes it more difficult for an attacker, since they do not have access to data and system resources simply by gaining access to the system.

With these concepts in mind, it is recommended to implement a secure coding policy prior to the start of development. This will include policies This also means that the secure coding policy will need to be evaluated and updated throughout development to keep it effective. For the best secure practices, implementing a zero trust framework is also recommended. The rise in the number of attacks and the amount and types of data being held encourage us to adopt more comprehensive security policies. Zero trust also helps security by planning for a failure of security. When a malicious user penetrates a system with a zero trust framework, they have less access than they would with a traditional security framework. Additional policies include making use of encryption and AAA standards. Encrypting data is another method to plan for failure, making it difficult to impossible for a malicious user to make use of data due to the encryption. AAA standards include authentication, authorization, and auditing. Authentication involves the verification of users and authorization is the process of verifying what data and resources the user has access to. Auditing is particularly important for identifying problems, as it’s the process of checking when users log in and log out, and what resources and data are accessed when. This can help identify if an attack is taking place if a user is access resources or data they should not have access to.