Module Eight

Eric Diaz

Southern New Hampshire University

CS-405

20 August 2025

Adopting a secure coding standard is very important, because if you leave security at the end it can take longer to debug and can lead to data leaks. You should plan the security features at the start, then implement it while coding so you can ensure that the security features are added to the right spot. Having to add security at the end is just a bad planning and coding practice. A good secure coding standard to adopt is the University of Michigan Secure Coding and Application Security standard. Some requirements are using a test environment, implementing two-factor authorization, and having proper error handling techniques.

Evaluating the risk of an attack to cost is important. The biggest threats we need to cover and make sure no damage is done. Compared to a threat that is unlikely to happen, we can save resources and allocate it somewhere else.

Zero trust is a security model and it does not trust the user at all. Not trusting the user at all is the best way to ensure no cyber attacks happen from the inside. Zero trust relies on continuous monitoring and validation of user identities, device security, and network behavior. Also having two-factor authorization like with a pin or fingerprint. This can affect the user by having them be annoyed with how often they have to verify it is them and not a threat. Users can potentially be accidentally locked out of their account if they mess up the signing process. Overall this is a great way to ensure your program is safe since there are security layers all throughout the system.

A security policy that I definitely recommend is validating input data which is crucial to ensuring a system is safe with handling data. It protects the program from SQL injections and cross-site scripting. There are many other security policy recommendations that should be followed, and ensuring your team follows it is crucial.