Shawn Plaisted

CS-405

Southern New Hampshire University

October 22, 2025

Module 8

One of the biggest lessons I have learned in this course is that security should never be saved for the end of a project. Building it in from the start makes everything stronger and easier to manage later on. Adopting a secure coding standard helps create a consistent way for developers to write safer code, and it sets the tone for the whole process. I used to think of security as something you add after everything else works, but now I understand it needs to be part of every stage of development. Writing code with security in mind from the beginning reduces mistakes and prevents vulnerabilities before they ever have a chance to appear.

Evaluating and assessing risk has also become a big part of how I look at security now. It is not just about finding problems but understanding how serious they are and how likely they are to happen. Every risk has a cost, and every fix does too, so balancing those decisions matters. Some risks can be reduced easily, while others might take more time or money to fix. Learning how to weigh the cost and benefit of mitigation has shown me that good security is about smart choices, not just strict rules. It is about managing risk in a realistic and effective way.

The zero trust idea has completely changed how I think about protecting systems. It means that nobody and nothing is automatically trusted, no matter where they are in the network. Everything needs to be verified before being allowed access. At first, that might seem extreme, but when I think about how many attacks come from inside networks or through stolen credentials, it makes perfect sense. Zero trust creates a safer environment by making sure every request and connection is legitimate. It encourages a mindset where caution and verification are normal rather than optional.

Strong security policies are what hold all of this together. They create structure and make sure that everyone understands their responsibilities. A good policy explains who can access what, how data is protected, and what steps to take when something goes wrong. I have learned that without clear policies, even secure code can become vulnerable if people do not follow safe practices. Going forward, I plan to keep using these lessons in my work by following secure coding standards, regularly assessing risks, applying zero trust principles, and supporting strong security policies that protect both users and systems.