Chandler Duhamel  
Southern New Hampshire University  
CS-405  
June 25, 2025

One of the key learnings for me throughout this course is the idea of adopting secure coding practices early in development. When security is left until the end of the development process, it creates issues that will be more difficult and costlier to fix than if the secure development standard was integrated from the very beginning. Just like you will be better off starting with standards like the SEI CERT or OWASP from the very beginning, you will be better off creating software that is prosthetic secure than patch secured. This relates back to the DevSecOps ideation we spoke about, let's integrate security into the pipeline instead of having it occur post-development.

I have also learned to analyze risk and consider mitigation strategies are more calculated ways. It is not feasible to fix every vulnerability, you need to weigh the likelihood and impact of each vulnerability. Utilizing a CVSS score to work out severity was insightful, and balancing the fix against the cost and time was practical. Not every threat will require remediation in the sprint, sometimes they will take documentation and acceptance in a future sprint (based on its risk profile and stakeholder need).

Zero trust changed how I conceptualized system design. Assuming that there is no user, device, or service that is trustworthy, we need to enforce controls at every level: identify, segment networks, least privilege, and monitor continuously. At first, it might seem overboard, but zero trust philosophy makes sense in the current security landscape: cloud-native systems and distributed systems have made the traditional perimeter a concept of the past.

Lastly, writing strong security policies is not bureaucratic nonsense--policies should be practical enforceable guidelines, which match what people do. It doesn’t matter if the strong policy is for password complexity, code review, or incident response, it is important that they are practical, specific, and flexible. I would suggest reviewing and rewriting those policies with team input.