Edwin Jones

CS-405: Secure Coding

June 19, 2025

Consider the Motive for the Attack

Understanding the underlying motive behind an attack is crucial when developing secure software. While the motive may not always be clear, applying a mindset that actively searches for one enables me to better anticipate possible threat vectors. In my own practice, I will routinely ask not just "can this be exploited?" but also "why would someone exploit this?" Whether the goal is financial gain, intellectual property theft, or reputational damage, identifying the incentive behind a potential attack can help prioritize mitigation strategies. This helps in applying secure coding principles more effectively—especially in sensitive areas such as input validation, access control, and data encryption.

When mentoring a new developer, I would explain this concept using a practical analogy: "Think like an attacker." I’d emphasize that every line of code, every feature exposed to the outside world, might represent a potential opportunity for someone with malicious intent. I would guide them to evaluate their code not just for correctness or performance, but from the perspective of someone trying to break or exploit it. Incorporating threat modeling into their workflow early on would be key to making this habit intrinsic to their development process.

An example I plan to use in my Module Eight reflection involves my work on buffer overflow and SQL injection exercises earlier in the course. In both cases, the vulnerabilities exist not because the developers failed to code functionality, but because they didn’t consider the motives behind attacks that would exploit unsafe memory or poorly sanitized inputs. By shifting focus toward understanding those motives, I’ve become more intentional about building security in from the start—not as an afterthought.