8-2 Journal: Portfolio Reflection

Colin Aheron

Security is an essential part of software development, and its adoption should be integrated from the start rather than treated as an afterthought. Following a secure coding standard ensures consistent practices across the team, helping to prevent common vulnerabilities like injection attacks or insecure data handling. By addressing security early in the design and development phases, teams can avoid costly fixes later and build more robust systems.

Evaluating risks and assessing the cost-benefit of mitigation strategies are critical for effective security. Developers and stakeholders must prioritize the most significant risks based on their likelihood and potential impact. For example, mitigating a vulnerability in authentication systems may take precedence over optimizing logging mechanisms. A balanced approach ensures resources are focused where they’re needed most.

The Zero Trust model plays a crucial role in modern security. Its principle of never trust, always verify ensures that every user, device, and connection is continuously authenticated and authorized. By treating all access as untrusted by default, Zero Trust significantly reduces the risk of insider threats, phishing attacks, and unauthorized lateral movement within a network.

Implementing effective security policies is key to maintaining a secure environment. These policies should include regular security training for developers, enforcing least privilege access, using multifactor authentication, and ensuring data is encrypted both in transit and at rest. Organizations should also adopt a defense-in-depth strategy, layering security measures to protect against a variety of threats. By embedding security into the development lifecycle, assessing risks intelligently, adopting Zero Trust principles, and enforcing clear security policies, developers and organizations can build systems that are resilient to modern threats.