Ali Alshara

CS 405

10/27/24

8-2 Journal

So what I had taken away from this course was instead of waiting to implement security measures for the end, incorporating security at every stage of the Software Development Lifecycle can significantly reduce associated risks. Establishing secure coding standards from the beginning is essential, with security considerations ideally addressed even before coding begins. By integrating Defense in Depth principles early, the potential impact of security risks is minimized.

Security threats can arise from both internal and external sources, making risk assessment a fundamental part of decision making. As threats evolve, it’s vital to address all potential risks; otherwise, the costs of mitigation can be much higher. Simple coding techniques, early and frequent testing, and consistent static code analysis following each amendment play a crucial role. Regular automated testing, including unit and penetration tests, further strengthens security.

With rapid advancements in technology, zero trust has become essential due to limitations in traditional authentication methods. Every user, device, network, and data access point must undergo verification. Implementing multifactor authentication, biometrics, and Single Sign On systems ensures that security is consistently upheld.

Security policies provide guidelines that help safeguard code and data, establishing core principles, coding standards, and best practices essential for development projects. By adhering to these policies, organizations ensure a consistent approach to maintaining security.

As technology accelerates, the FTC’s lessons learned offer valuable insights into common vulnerabilities. Key practices, like controlling data access, securing authentication, and safeguarding data storage and transmission, are essential. Additional measures, such as applying timely software patches and enforcing strong password requirements, help maintain a secure environment.