Alexander Ray 10/26/2025

* Adoption of a secure coding standard, and not leaving security to the end

Adoption of a Secure Coding Standard addresses how you build defenses based on your risk assessment. Adopting a secure coding standard (like the OWASP Top 10 or CERT Secure Coding Standards) is the most effective way to prevent vulnerabilities from being introduced in the first place. The key here is "not leaving security to the end." This is the core principle of DevSecOps security.

* Evaluation and assessment of risk and cost benefit of mitigation

Evaluation and Assessment of Risk and Cost-Benefit of Mitigation is the essential starting point for any intelligent security program. You cannot protect everything equally, and you shouldn't try. A proper risk assessment is the process of identifying your most valuable assets ( like user data, financial records, intellectual property) and then analyzing the likelihood and potential impact of threats against them. The cost benefit analysis is a critical business facing component. Security measures are not free they cost time, money, and developer resources, and they can add friction for users.

* Zero trust

Zero Trust is the guiding philosophy that informs both your policies and your coding standards. It's a complete rejection of the old security model, which assumed anything inside the network was safe. The guiding sentiment is "Never trust, always verify."

Zero Trust assumes the network is already compromised. Trust is never granted implicitly.

* Implementation and recommendations of security policies

Security policies are the high-level, formal rules that define an organization’s strategic approach to protecting its systems, data, and users. A good security policy is living and practical. It evolves with the threat landscape and organizational changes, and it’s supported by automation where possible for example, integrating policy checks into CI/CD pipelines or access control systems. Regular reviews, metrics, and feedback loops ensure it stays relevant, however, implementation is the tricky part. A policy that exists only as a document has little real value if it isn’t integrated into daily operations. Policies must be communicated, enforced, and operationalized through clear procedures, training, and tooling. Developers should understand how a policy affects their code reviews and build processes, IT staff should see how it maps to configuration baselines and monitoring; and management should track compliance and effectiveness.